



**Escuela de Doctorado  
y Estudios de Posgrado**  
Universidad de La Laguna

ESCUELA DE DOCTORADO Y  
ESTUDIOS DE POSTGRADO

## **Trabajo de Fin de Máster**

### **Desarrollo de un BMS y su batería para sistemas aislados**

**Titulación:** Máster en Ingeniería Industrial

#### **Estudiante:**

Augusto Samuel Hernández Martín

**Tutor:** Fernando Luis Rosa González

**Co-Tutor:** Iván Rodríguez Méndez

5 de septiembre de 2021

La publicación de este Trabajo Fin de Máster solo implica que el estudiante ha obtenido al menos la nota mínima exigida para superar la asignatura correspondiente, no presupone que su contenido sea correcto, aunque si aplicable. En este sentido, la ULL no posee ningún tipo de responsabilidad hacia terceros por la aplicación total o parcial de los resultados obtenidos en este trabajo. También pone en conocimiento del lector que, según la ley de protección intelectual, los resultados son propiedad intelectual del alumno, siempre y cuando se haya procedido a los registros de propiedad intelectual o solicitud de patentes correspondientes con fecha anterior a su publicación.



**IMPRESO DE AUTORIZACIÓN DEL  
TRABAJO DE FIN DE MÁSTER POR EL  
TUTOR**

**Curso 2020/2021**

---

El Dr. D. **Fernando Luis Rosa González**, con D.N.I. 43611314-W, como tutor del estudiante D. **Augusto Samuel Hernández Martín** en el Trabajo de Fin de Máster titulado

**Desarrollo de un BMS y su batería para sistemas aislados,**

da su autorización, acreditada por la firma electrónica de este documento, para la presentación y defensa de dicho proyecto, a la vez que confirma que el estudiante ha cumplido con los objetivos generales y particulares que lleva consigo la realización del mismo.

La Laguna, 5 de septiembre de 2021.

## **Agradecimientos**

Gracias a mis tutores Fernando Rosa e Iván Rodríguez por la dedicación y el apoyo prestado durante este proyecto. Agradecer la libertad concedida, fruto de la confianza depositada en mí. Agradecer a Delfín Díaz, Carmelo Militelo y el equipo de *ULL Dynamics* por sus consejos y cederme la herramienta necesaria para la ejecución de este proyecto. Gracias al resto de profesores que han impartido docencia en este máster, porque cada uno a su forma me ha aportado, además de los conocimientos propios, un punto de vista diferente, me han ayudado a ser crítico y mejorar la toma de decisiones.

A mis familiares, en especial a mis padres y mi hermana Silvia. A Priscila, gracias por estar ahí día a día siendo un apoyo fundamental. Gracias por soportar que cada día mi único tema de conversación fuese el mismo, por aguantar mis quejas sobre los problemas sobrevenidos así como tu apoyo para solucionarlos. Gracias por entenderme y sacarme una sonrisa hasta en los momentos difíciles. Sin tu comprensión este proyecto no habría recibido tantas horas de dedicación.

No podría olvidarme de mis compañeros de Máster. Gracias por los momentos vividos durante estos dos años y todo lo que he aprendido de ustedes.

Muchísimas gracias a todos.





**Escuela de Doctorado  
y Estudios de Posgrado**  
Universidad de La Laguna

ESCUELA DE ESTUDIOS DE  
DOCTORADO Y POSTGRADO

## **Trabajo de Fin de Master**

**Desarrollo de un BMS y su batería para sistemas  
aislados**

**TOMO I**

**Memoria**

**Titulación:** Máster en Ingeniería Industrial

**Estudiante:**

Augusto Samuel Hernández Martín

**Tutor:** Fernando Luis Rosa González

**Co-Tutor:** Iván Rodríguez Méndez

5 de septiembre de 2021

## Resumen

El presente Trabajo Fin de Máster consiste en el diseñado y fabricación de un *Sistema de Gestión de Batería (BMS)*, así como una batería modular configurable a diferentes tensiones y capacidades, con el objetivo de poder alimentar sistemas electrónicos aislados como los dispositivos de capturas de índices de significación eco-acústicos. Se ha llevado a cabo el diseño electrónico de los diferentes módulos que forman el sistema de alimentación y gestión así como el cálculo y diseño de la batería a partir de celdas de ion-litio y su estructura mecánica. Para poder alimentar dicho sistema se ha dimensionado una posible instalación generadora, empleando energía fotovoltaica. Finalmente, se ha realizado la fabricación de los distintos elementos, tanto mecánicos como electrónicos, y el desarrollo *software* de las diferentes herramientas necesarias para la configuración del BMS, la captura y el análisis de los datos generados. Por último, se han propuesto y analizado diferentes aplicaciones, además de la mencionada como objetivo principal del proyecto, para este sistema de BMS y acumulador.

## ***Abstract***

*In the present Master's Degree Final Project, a Battery Management System (BMS) has been designed and manufactured, as well as a modular battery configurable at different voltages and capacities, with the aim of being able to power isolated electronic systems such as devices for capturing eco-acoustic significance indexes. The electronic design of the different modules that make up the power supply and management system has been carried out, as well as the calculation and design of the battery based on lithium-ion cells and its mechanical structure. In order to be able to supply this system, a possible generating installation has been dimensioned, using photovoltaic energy. Finally, the manufacture of the different elements, both mechanical and electronic, and the software development of the different tools necessary for the configuration of the BMS and the capture and analysis of the data generated have been carried out. Finally, different applications have been proposed and analysed, in addition to the one mentioned as the main objective of the project, for this BMS and accumulator system.*



# Índice general

<b>I Memoria</b>	<b>7</b>
<b>Resumen</b>	<b>8</b>
<b>Abstract</b>	<b>9</b>
<b>1. Introducción</b>	<b>27</b>
1.1. Antecedentes . . . . .	27
1.2. Objetivos . . . . .	28
1.3. Cronología del desarrollo . . . . .	29
1.4. Distribución de la memoria . . . . .	30
<b>2. Material y método</b>	<b>35</b>
2.1. Material . . . . .	35
2.1.1. Distribución Linux . . . . .	35
2.1.2. <i>Git</i> . . . . .	36
2.1.3. <i>KiCAD EDA</i> . . . . .	36
2.1.4. <i>FreeCAD</i> . . . . .	37
2.1.5. <i>Inkscape</i> . . . . .	38
2.1.6. <i>FlatCAM</i> . . . . .	38
2.1.7. <i>Prusa Slic3r</i> . . . . .	38
2.1.8. <i>Python</i> y <i>C++</i> . . . . .	39
2.1.9. <i>QtDesigner</i> y <i>PyQT</i> . . . . .	39
2.1.10. <i>Atom</i> y <i>Platformio</i> . . . . .	39
2.1.11. $\text{\LaTeX}$ y <i>TeXstudio</i> . . . . .	39
2.1.12. Máquina CNC-4030Z . . . . .	40
2.1.13. Horno de soldadura . . . . .	40
2.1.14. Impresoras 3D . . . . .	41
2.1.15. Lupa electrónica . . . . .	41
2.1.16. Estación de soldadura por punto . . . . .	42
2.1.17. Carga electrónica Rigol DL3031A . . . . .	42
2.2. Método . . . . .	43

<b>3. Diseño e integración del sistema</b>	<b>45</b>
3.1. Descripción general del sistema . . . . .	45
3.2. Descripción de los elementos del sistema . . . . .	46
3.2.1. Sistema de Monitoreo de Baterías (BMS) . . . . .	46
3.2.2. Configurador de BMS <i>CAN2Serial</i> . . . . .	51
3.2.3. Programador ICSP . . . . .	51
3.2.4. Batería . . . . .	52
3.2.5. Cargador solar MPPT . . . . .	55
<b>4. Diseño electrónico</b>	<b>57</b>
4.1. Módulo BMS . . . . .	57
4.1.1. Microcontrolador ATmega 328P . . . . .	57
4.1.2. Circuito Integrado de Monitoreo LTC6804 . . . . .	59
4.1.3. Multiplexado y lectura de los termistores . . . . .	64
4.1.4. Periféricos de comunicación CAN BUS . . . . .	66
4.1.5. Interfaces de Entrada y Salida del BMS . . . . .	67
4.1.6. Diseño del circuito impreso . . . . .	68
4.1.7. Modelo 3D del circuito impreso . . . . .	70
4.2. Módulo conversor CAN2Serial . . . . .	71
4.2.1. Microcontrolador ATmega 328P . . . . .	71
4.2.2. Periféricos de comunicación CAN BUS . . . . .	72
4.2.3. Comunicación RS232 . . . . .	72
4.2.4. Interfaces de Entrada/Salida . . . . .	74
4.2.5. Diseño del circuito impreso . . . . .	74
4.2.6. Modelo 3D del circuito impreso . . . . .	75
4.3. Módulo conversor RS485 EPSolar-CAN . . . . .	75
4.3.1. Comunicación RS485 . . . . .	76
4.3.2. Alimentación del módulo . . . . .	77
4.3.3. Interfaces de Entrada/Salida . . . . .	78
4.3.4. Diseño del circuito impreso . . . . .	78
4.3.5. Modelo 3D del circuito impreso . . . . .	79
4.4. Programador ICSP . . . . .	79
4.4.1. Modelo 3D del circuito impreso . . . . .	80
<b>5. Fabricación de PCBs</b>	<b>83</b>
5.1. Características físicas de las placas . . . . .	83
5.2. Metodología de fabricación de PCBs . . . . .	84
5.2.1. Fabricación de placas de circuito impreso con CNC . . . . .	84
5.2.2. Fabricación de placas de circuito impreso con métodos químicos . . . . .	85
5.2.3. Soldadura de las placas de circuito impreso . . . . .	86

5.3.	Fabricación del módulo CAN2Serial . . . . .	86
5.3.1.	Proceso de fabricación de la PCB . . . . .	87
5.3.2.	Soldadura de los componentes . . . . .	88
5.3.3.	Resultado final . . . . .	90
<b>6.</b>	<b>Diseño y Fabricación de la batería</b> . . . . .	<b>91</b>
6.1.	Diseño de la batería . . . . .	91
6.1.1.	Caracterización del sistema . . . . .	91
6.1.2.	Diseño mecánico de la batería . . . . .	100
6.1.3.	Aspecto de la batería de 4 paquetes en serie . . . . .	104
6.1.4.	Diseño de la caja de la batería . . . . .	104
6.2.	Fabricación de la batería . . . . .	106
6.2.1.	Características mecánicas de las piezas . . . . .	106
6.2.2.	Metodología de fabricación de piezas . . . . .	107
6.2.3.	Fabricación de los soportes de las celdas . . . . .	108
6.2.4.	Fabricación de las láminas de unión de las celdas . . . . .	108
6.2.5.	Montaje y soldadura de las pletinas . . . . .	109
6.2.6.	Preparación del cableado . . . . .	109
6.2.7.	Fabricación de la caja protectora de la batería . . . . .	111
6.3.	Montaje y fabricación de la carcasa del CAN2Serial . . . . .	112
6.4.	Montaje y fabricación del <i>configurador portátil</i> . . . . .	113
<b>7.</b>	<b>Cálculo del sistema de generación</b> . . . . .	<b>115</b>
7.1.	Premisas de cálculo . . . . .	115
7.2.	Consumo diario y determinación del ángulo de inclinación . . . . .	115
7.3.	Dimensionado del generador fotovoltaico . . . . .	117
7.4.	Dimensionado del regulador de carga . . . . .	118
7.5.	Dimensionado de las protecciones . . . . .	118
7.6.	Simulación de los cálculos . . . . .	119
7.7.	Esquema de instalación . . . . .	120
<b>8.</b>	<b>Desarrollo de <i>Software</i></b> . . . . .	<b>123</b>
8.1.	Módulo BMS . . . . .	123
8.1.1.	Comunicación con el LTC6804 . . . . .	125
8.1.2.	Lectura de temperaturas y corriente . . . . .	128
8.1.3.	Almacenamiento de parámetros . . . . .	130
8.1.4.	Estimación del SOC . . . . .	133
8.1.5.	Comunicación CAN BUS . . . . .	134
8.2.	Módulo CAN2Serial . . . . .	142
8.2.1.	Mensajes Serial a enviar . . . . .	143
8.2.2.	Interfaz gráfica de configuración . . . . .	147

8.3.	Herramienta de procesamiento de datos BMS . . . . .	150
8.3.1.	Código implementado . . . . .	150
8.4.	<i>Software</i> SOC Estimation . . . . .	155
8.5.	<i>Software</i> de la Herramienta de cálculo de consumos del nodo . . .	155
8.5.1.	Capturadora de consumos en el nodo . . . . .	156
8.5.2.	Herramienta de representación de datos . . . . .	158
<b>9.</b>	<b>Pruebas del Sistema</b>	<b>161</b>
9.1.	Metodología de las pruebas realizadas . . . . .	161
9.2.	Prueba de descarga a 44 $\Omega$ . . . . .	162
9.3.	Prueba de descarga a 11 $\Omega$ . . . . .	167
9.4.	Prueba de descarga a 10 A . . . . .	172
9.5.	Prueba de carga a 15A . . . . .	178
9.6.	Pruebas de comunicación CAN . . . . .	183
<b>10.</b>	<b>Configuración y modos de uso</b>	<b>187</b>
10.1.	Conexión de la batería y BMS . . . . .	187
10.2.	Conexión del CAN2Serial y <i>BMS Programmer</i> . . . . .	187
<b>11.</b>	<b>Aplicaciones y Casos de uso</b>	<b>193</b>
11.1.	Aplicación en vehículos eléctricos . . . . .	193
11.1.1.	Aplicación del prototipo al proyecto <i>ULL Dynamics</i> . . .	194
11.2.	Aplicación en vehículos no tripulados . . . . .	196
	<b>Conclusiones</b>	<b>199</b>
	<b>Conclusions</b>	<b>201</b>
<b>II</b>	<b>Pliego de condiciones y presupuesto</b>	<b>207</b>
<b>12.</b>	<b>Pliego de condiciones</b>	<b>209</b>
<b>13.</b>	<b>Presupuesto del Proyecto</b>	<b>211</b>
<b>III</b>	<b>Anexos</b>	<b>215</b>
<b>A.</b>	<b>Cronología del Proyecto</b>	<b>217</b>
A.1.	Diagrama de Gantt . . . . .	218
A.2.	Cuadro de Temporalización . . . . .	219

<b>B. Gráficos caracterización de las celdas</b>	<b>221</b>
B.1. Curva de SOC de la celda . . . . .	222
B.2. Regresión polinómica de la curva SOC . . . . .	223
B.3. Regresión polinómica de la curva SOC. Grado 2 . . . . .	224
B.4. Regresión polinómica de la curva SOC. Grado 3 . . . . .	225
B.5. Regresión polinómica de la curva SOC. Grado 4 . . . . .	226
B.6. Regresión polinómica de la curva SOC. Grado 5 . . . . .	227
B.7. Regresión polinómica de la curva SOC. Grado 6 . . . . .	228
<b>C. Gráficos de consumo del nodo</b>	<b>229</b>
C.1. Consumo de corriente del día 21 de abril de 2021 . . . . .	230
C.2. Consumo de potencia del día 21 de abril de 2021 . . . . .	231
C.3. Voltaje del sistema el día 21 de abril de 2021 . . . . .	232
C.4. Temperatura del sistema el día 21 de abril de 2021 . . . . .	233
C.5. Consumo de corriente del día 22 de abril de 2021 . . . . .	234
C.6. Consumo de potencia del día 22 de abril de 2021 . . . . .	235
C.7. Voltaje del sistema el día 22 de abril de 2021 . . . . .	236
C.8. Temperatura del sistema el día 22 de abril de 2021 . . . . .	237
<b>D. Gráficos de tests del sistema</b>	<b>239</b>
D.1. Corriente extraída de la batería. <i>Test</i> de 44 $\Omega$ . . . . .	240
D.2. Tensión de las celdas. <i>Test</i> de 44 $\Omega$ . . . . .	241
D.3. Diferencia de tensión de las celdas. <i>Test</i> de 44 $\Omega$ . . . . .	242
D.4. SOC de la batería. <i>Test</i> de 44 $\Omega$ . . . . .	243
D.5. Temperatura de la batería. <i>Test</i> de 44 $\Omega$ . . . . .	244
D.6. Corriente extraída de la batería. <i>Test</i> de 11 $\Omega$ . . . . .	245
D.7. Tensión de las celdas. <i>Test</i> de 11 $\Omega$ . . . . .	246
D.8. Diferencia de tensión de las celdas. <i>Test</i> de 11 $\Omega$ . . . . .	247
D.9. SOC de la batería. <i>Test</i> de 11 $\Omega$ . . . . .	248
D.10. Temperatura de la batería. <i>Test</i> de 11 $\Omega$ . . . . .	249
D.11. Corriente extraída de la batería. <i>Test</i> de 10 A . . . . .	250
D.12. Tensión de las celdas. <i>Test</i> de 10 A . . . . .	251
D.13. Diferencia de tensión de las celdas. <i>Test</i> de 10 A . . . . .	252
D.14. SOC de la batería. <i>Test</i> de 10 A . . . . .	253
D.15. Tensión de la batería. <i>Test</i> de 10 A . . . . .	254
D.16. Temperatura de la batería. <i>Test</i> de 10 A . . . . .	255
D.17. Corriente de carga a la batería. <i>Test</i> de carga a 15 A . . . . .	256
D.18. Tensión de las celdas. <i>Test</i> de carga a 15 A . . . . .	257
D.19. Diferencia de tensión de las celdas. <i>Test</i> de carga 15 A . . . . .	258
D.20. SOC de la batería. <i>Test</i> de carga a 15 A . . . . .	259
D.21. Tensión de la batería. <i>Test</i> de carga a 15 A . . . . .	260

D.22. Temperatura de la batería. <i>Test</i> de 10 A . . . . .	261
<b>E. <i>Datasheets</i> de los Integrados y componentes utilizados</b>	<b>263</b>
E.1. LTC6804 . . . . .	264
E.2. ACS712 . . . . .	265
E.3. BSS308PE . . . . .	266
E.4. ADG731 . . . . .	267
E.5. ATmega 328P . . . . .	268
E.6. MAX232 . . . . .	269
E.7. MAX485 . . . . .	270
E.8. MCP2515 . . . . .	271
E.9. TJA1050T . . . . .	272
E.10. HC-49 Oscilador . . . . .	273
E.11. INA 219 . . . . .	274
E.12. MP 2307 DC/DC . . . . .	275
E.13. Resistencia 10 $\Omega$ . . . . .	276
E.14. Conector RJ45 . . . . .	277
E.15. Conector JST-XH 14 . . . . .	278
E.16. Conector KF2510 2 pines . . . . .	279
E.17. Conector IP68 WP3/EH . . . . .	280
E.18. Portafusible 5mm x 20mm . . . . .	281
E.19. Raspberry Pi 3B+ . . . . .	282
<b>F. Diseño eléctrico de la batería</b>	<b>283</b>
F.1. <i>Datasheet</i> de la celda INR21700 . . . . .	284
F.2. Chapa de Cobre de 0.8 mm . . . . .	285
<b>G. Dimensionado y simulación del generador fotovoltaico</b>	<b>287</b>
G.1. Cálculo del generador . . . . .	288
G.2. Resultado de las simulaciones . . . . .	289
<b>H. <i>Datasheet</i> del sistema de generación</b>	<b>291</b>
H.1. Panel fotovoltaico Xunzel PowerSolar 60Wp . . . . .	292
H.2. Cargador MPPT 3210AN . . . . .	293
<b>I. Hojas de esquema del diseño electrónico</b>	<b>295</b>
<b>J. Planos de diseño de placas de circuito impreso</b>	<b>301</b>
<b>K. Modelos 3D de los módulos desarrollados</b>	<b>319</b>

<i>ÍNDICE GENERAL</i>	17
<b>L. Diagramas de flujo del <i>firmware</i> desarrollado</b>	<b>329</b>
L.1. Diagrama de flujo del BMS . . . . .	330
L.2. Diagrama de flujo de CAN2Serial . . . . .	331
<b>M. Planos de diseño de la batería y herramientas</b>	<b>333</b>



# Índice de figuras

2.1. CNC-4030Z. . . . .	40
2.2. Horno de soldadura T962. . . . .	41
2.3. Lupa e Impresora 3D empleadas. . . . .	42
2.4. Estación de soldadura por punto empleada. . . . .	43
2.5. Carga electrónica Rigol DL3031A. . . . .	43
3.1. Diagrama de bloques general del sistema. . . . .	46
3.2. Diagrama de bloques del BMS. . . . .	47
3.3. Diagrama de bloques de funcionamiento del microcontrolador. . . . .	48
3.4. Esquema de conexión recomendado por el fabricante.[45, pág. 78]	49
3.5. Ejemplo de implementación de un sistema con MCP2515. . . . .	51
3.6. Diseño 3D de la interfaz de comunicación CAN2Serial. . . . .	51
3.7. Diseño 3D del programador ICSP desarrollado. . . . .	52
3.8. Aspecto final de la batería diseñada. . . . .	53
3.9. Celda Samsung INR21700-40T empleada. . . . .	54
3.10. Cargador solar y sistema de comunicación. . . . .	55
4.1. Esquema de conexiones del ATmega 328P del BMS (Anexo I). . . . .	58
4.2. Conectores de E/S del ATmega 328P del BMS (Anexo I). . . . .	59
4.3. Filtrado en la alimentación del ATmega 328P del BMS (Anexo I). . . . .	59
4.4. Esquema de conexión del LTC6804 (Anexo I). . . . .	60
4.5. Esquema de selección de la alimentación del LTC6804 (Anexo I). . . . .	60
4.6. Regulador lineal de 5V propuesto por Analog Devices[45]. . . . .	61
4.7. Tabla de configuración del DCTO[45]. . . . .	61
4.8. Filtro paso-bajo para medición de la tensión de las celdas (Anexo I). . . . .	62
4.9. Circuito de balanceo controlado por un MOSFET de canal P (Anexo I). . . . .	63
4.10. Gráfico de resistencia de conducción según $I_D$ y $V_{GS}$ [25, pág. 5]. . . . .	63
4.11. Divisor empleado para la lectura de las NTC. . . . .	64
4.12. Esquema de conexión del multiplexor (Anexo I). . . . .	65
4.13. Esquema de conexión del MCP2515 y TJA1050 (Anexo I). . . . .	66

4.14. Esquema de conectores del BMS (Anexo I). . . . .	68
4.15. Distribución de componentes en la PCB del BMS. . . . .	69
4.16. Ruteado de pistas del BMS. . . . .	70
4.17. Modelos 3D del BMS. . . . .	71
4.18. Conexión del ATmega 328P del módulo CAN2Serial (Anexo I). . . . .	72
4.19. Conexión de los periféricos CAN de CAN2Serial (Anexo I). . . . .	73
4.20. Conexión del MAX232 del módulo CAN2Serial (Anexo I). . . . .	73
4.21. Interfaces de Entrada y Salida de CAN2Serial (Anexo I). . . . .	74
4.22. Pincado de los conectores DB9. . . . .	74
4.23. Ruteado de pistas del CAN2Serial. . . . .	75
4.24. Modelos 3D del CAN2Serial. . . . .	76
4.25. Conexión del MAX485 en el módulo EPSolar-CAN (Anexo I). . . . .	77
4.26. Circuitos de alimentación del convertidor EPSolar-CAN (Anexo I). . . . .	77
4.27. Interfaces E/S del módulo EPSolar-CAN (Anexo I). . . . .	78
4.28. Ruteado de pistas del módulo EPSolar-CAN. . . . .	79
4.29. Modelos 3D del módulo EPSolar-CAN. . . . .	79
4.30. Esquema de conexión del programador ICSP (Anexo I). . . . .	80
4.31. Ruteado de pistas del programador ICSP. . . . .	81
4.32. Modelos 3D del Programador ICSP. . . . .	81
5.1. Etapas del proceso de fabricación. . . . .	87
5.2. Atacado químico y revisión de la PCB. . . . .	87
5.3. Componentes SMD colocados. . . . .	88
5.4. Resultado de la PCB tras soldar los componentes SMD. . . . .	89
5.5. Componentes THT soldados mediante soldadura manual. . . . .	89
5.6. Acabado final de las PCBs. . . . .	90
6.1. Nodo de captura de datos <i>SmartNode</i> fabricado. . . . .	92
6.2. Consumos de corriente del sistema (ver Anexo C.1 y C.5). . . . .	93
6.3. Tensión del sistema (ver Anexo C.3 y C.7). . . . .	93
6.4. Consumo de potencia del sistema (ver Anexo C.2 y C.6). . . . .	94
6.5. Esquema de configuración de la batería (ver Anexo F.2). . . . .	95
6.6. Circuitos equivalentes de la batería. . . . .	96
6.7. Curva del SOC (Anexo B.1). . . . .	98
6.8. Curva del SOC empírica y calculada con grado 4 (Anexo B.5). . . . .	99
6.9. Estructura básica de un <i>paquete</i> . . . . .	100
6.10. Tapas laterales de los paquetes . . . . .	101
6.11. Aspecto del paquete de celdas diseñado. . . . .	104
6.12. Aspecto de la base de la caja de la batería. . . . .	105
6.13. Aspecto de la caja de la batería. . . . .	106
6.14. Módulo de 1 paquete de batería impreso. . . . .	108

6.15. Pletinas de cobre-níquel fabricadas. . . . .	109
6.16. Proceso de soldadura de 1 paquete de batería. . . . .	110
6.17. Batería completa montada con su cableado. . . . .	110
6.18. Batería montada con el BMS. . . . .	111
6.19. Piezas impresas de la base de la caja de batería. . . . .	111
6.20. Montaje de la caja de la batería. . . . .	112
6.21. Carcasa de CAN2Serial diseñada. . . . .	112
6.22. Carcasa de CAN2Serial fabricada. . . . .	113
6.23. Carcasa del <i>configurador</i> fabricada. . . . .	113
7.1. Captura de los datos introducidos en el simulador. . . . .	119
7.2. Probabilidad del SOC de la batería al finalizar el día (ver Anexo G.2). . . . .	120
7.3. Esquema de instalación del sistema adoptado. . . . .	120
8.1. Diagrama de flujo del <i>firmware</i> del BMS (Anexo L.1). . . . .	124
8.2. Tabla de los Registros de Configuración[45]. . . . .	126
8.3. Tabla de Registros de las Celdas C1 a C3[45]. . . . .	126
8.4. Trama del mensaje SPI del ADG731[2]. . . . .	130
8.5. Trama del mensaje CAN de voltaje mínimo. . . . .	134
8.6. Trama del mensaje CAN de voltaje máximo. . . . .	135
8.7. Trama del mensaje CAN de DCTO. . . . .	135
8.8. Trama del mensaje CAN de configuración de celdas en serie. . . . .	135
8.9. Trama del mensaje CAN de configuración de celdas en paralelo. . . . .	136
8.10. Trama del mensaje CAN de configuración del número de terminales. . . . .	136
8.11. Trama del mensaje CAN de configuración del tiempo entre mensajes. . . . .	137
8.12. Trama del mensaje CAN para forzar el balanceo de las Celdas 1 a 8. . . . .	137
8.13. Trama del mensaje CAN para forzar el balanceo de las Celdas 9 a 12. . . . .	137
8.14. Trama del mensaje CAN para configurar cuando realizar el balanceo. . . . .	138
8.15. Trama del mensaje CAN para configurar la máxima diferencia entre celdas. . . . .	138
8.16. Trama del mensaje CAN para configurar el offset del sensor de corriente. . . . .	139
8.17. Trama del mensaje CAN para pedir las configuraciones del BMS. . . . .	139
8.18. Mensajes CAN de respuesta con la configuración del BMS. . . . .	140
8.19. Mensajes CAN con las tensiones de las celdas. . . . .	140
8.20. Mensajes CAN con las temperaturas medidas. . . . .	141
8.21. Trama del mensaje CAN con el Estado de Carga. . . . .	141

8.22. Trama del mensaje CAN con el Estado de Salud. . . . .	141
8.23. Trama del mensaje CAN con la corriente de la batería. . . . .	142
8.24. Flujograma del <i>firmware</i> del módulo CAN2Serial (Anexo L.2). . .	143
8.25. Capturas del <i>software</i> BMS Programmer. . . . .	149
8.26. Ejemplos de gráficas generadas con los datos del BMS. . . . .	154
8.27. Salida por consola de la aproximación polinómica. . . . .	155
8.28. Fragmento del fichero <i>json</i> generado. . . . .	156
8.29. Estructura de ficheros generada. . . . .	157
8.30. Estructura de ficheros generada. . . . .	159
9.1. Esquema del montaje adoptado para las pruebas de carga y descarga.	162
9.2. Corriente de descarga medida en el <i>test</i> de 44Ω (Anexo D.1). . . .	162
9.3. Voltajes de las celdas medidos en el <i>test</i> de 44Ω (Anexo D.2). . .	163
9.4. Diferencia máxima entre celdas durante el <i>test</i> de 44Ω (Anexo D.3).	164
9.5. Estado de Carga de la batería durante el <i>test</i> de 44Ω (Anexo D.4).	165
9.6. Temperaturas medidas durante el <i>test</i> de 44Ω (Anexo D.5). . . . .	166
9.7. Corriente de descarga medida en el <i>test</i> de 11Ω (Anexo D.6). . . .	167
9.8. Voltajes de las celdas medidos en el <i>test</i> de 11Ω (Anexo D.7). . .	168
9.9. Diferencia máxima entre celdas durante el <i>test</i> de 11Ω (Anexo D.8).	169
9.10. Estado de carga durante el <i>test</i> de 11Ω (Anexo D.9). . . . .	170
9.11. Temperaturas medidas durante el <i>test</i> de 11Ω (Anexo D.10). . . .	171
9.12. Corriente de descarga medida en el <i>test</i> a 10A (Anexo D.11). . . .	172
9.13. Voltajes de las celdas medidos en el <i>test</i> a 10A (Anexo D.12). . .	173
9.14. Voltajes de la batería en el <i>test</i> a 10A (Anexo D.15). . . . .	174
9.15. Diferencia máxima entre celdas durante el <i>test</i> de 10A (Anexo D.13). . . . .	175
9.16. SOC durante el <i>test</i> de 10A (Anexo D.14). . . . .	176
9.17. Temperaturas medidas durante el <i>test</i> de 10A (Anexo D.16). . . .	177
9.18. Corriente de carga medida en el <i>test</i> a 15A (Anexo D.17). . . . .	178
9.19. Voltajes de las celdas medidos en el <i>test</i> a 15A (Anexo D.18). . . .	179
9.20. Voltajes de la batería en el <i>test</i> a 15A (Anexo D.21). . . . .	180
9.21. Diferencia máxima entre celdas durante el <i>test</i> de 15A (Anexo D.19). . . . .	181
9.22. Estado de carga durante el <i>test</i> de 15A (Anexo D.20). . . . .	182
9.23. Temperaturas medidas durante el <i>test</i> de 15A (Anexo D.22). . . .	183
9.24. Captura con el tiempo transcurrido entre mensajes CAN. . . . .	184
9.25. Captura con el mensaje CAN para pedir configuraciones. . . . .	184
9.26. Captura con las respuestas CAN con las configuraciones. . . . .	184
9.27. Captura de mensajes CAN transmitidos por el bus. . . . .	185
9.28. Captura de mensajes CAN transmitidos por el bus. . . . .	185

10.1. Esquema de conexión del BMS. . . . .	188
10.2. <i>Configurador</i> y batería conectados. . . . .	188
10.3. <i>Configurador</i> ejecutando el <i>software</i> de configuración. . . . .	189
10.4. Pestaña <i>BMS Config</i> del <i>BMS Programmer</i> . . . . .	189
10.5. Pestaña <i>MainView</i> del <i>BMS Programmer</i> . . . . .	190
11.1. Batería en un vehículo eléctrico[55]. . . . .	193
11.2. Esquema propuesto para una batería con más de 12 celdas serie. . . . .	194
11.3. Proyectos vigentes de <i>ULL Dynamics</i> . . . . .	195
11.4. Estructura de un <i>glider</i> [48]. . . . .	196
B.1. Curva generada tras el ensayo de descarga de la celda. . . . .	222
B.2. Conjunto de curvas de regresión polinómica generadas. . . . .	223
B.3. Curva de regresión del SOC de grado 2. . . . .	224
B.4. Curva de regresión del SOC de grado 3. . . . .	225
B.5. Curva de regresión del SOC de grado 4. . . . .	226
B.6. Curva de regresión del SOC de grado 5. . . . .	227
B.7. Curva de regresión del SOC de grado 6. . . . .	228
C.1. Consumo de corriente del sistema <i>SmartNode</i> el 21/04/21. . . . .	230
C.2. Consumo de potencia del sistema <i>SmartNode</i> el 21/04/21. . . . .	231
C.3. Voltaje del sistema <i>SmartNode</i> el 21/04/21. . . . .	232
C.4. Temperatura del sistema <i>SmartNode</i> el 21/04/21. . . . .	233
C.5. Consumo de corriente del sistema <i>SmartNode</i> el 22/04/21. . . . .	234
C.6. Consumo de potencia del sistema <i>SmartNode</i> el 22/04/21. . . . .	235
C.7. Voltaje del sistema <i>SmartNode</i> el 22/04/21. . . . .	236
C.8. Temperatura del sistema <i>SmartNode</i> el 22/04/21. . . . .	237
D.1. Consumo de corriente de la batería con 44 $\Omega$ . . . . .	240
D.2. Tensión de las celdas con <i>test</i> de 44 $\Omega$ . . . . .	241
D.3. Diferencia de tensión de las celdas con <i>test</i> de 44 $\Omega$ . . . . .	242
D.4. SOC de la batería con <i>test</i> de 44 $\Omega$ . . . . .	243
D.5. Temperatura de la batería con <i>test</i> de 44 $\Omega$ . . . . .	244
D.6. Consumo de corriente de la batería con 11 $\Omega$ . . . . .	245
D.7. Tensión de las celdas con <i>test</i> de 11 $\Omega$ . . . . .	246
D.8. Diferencia de tensión de las celdas con <i>test</i> de 11 $\Omega$ . . . . .	247
D.9. SOC de la batería con <i>test</i> de 11 $\Omega$ . . . . .	248
D.10. Temperatura de la batería con <i>test</i> de 11 $\Omega$ . . . . .	249
D.11. Consumo de corriente de la batería con 10 A. . . . .	250
D.12. Tensión de las celdas con <i>test</i> de 10 A. . . . .	251
D.13. Diferencia de tensión de las celdas con <i>test</i> de 10 A. . . . .	252
D.14. SOC de la batería con <i>test</i> de 10 A. . . . .	253

D.15. Tensión de la batería con <i>test</i> de 10 A. . . . .	254
D.16. Temperatura de la batería con <i>test</i> de 10 A. . . . .	255
D.17. Corriente de carga de la batería con 15 A. . . . .	256
D.18. Tensión de las celdas con <i>test</i> de 15 A. . . . .	257
D.19. Diferencia de tensión de las celdas con <i>test</i> de 15 A. . . . .	258
D.20. SOC de la batería con <i>test</i> de 15 A. . . . .	259
D.21. Tensión de la batería con <i>test</i> de 15 A. . . . .	260
D.22. Temperatura de la batería con <i>test</i> de 15 A. . . . .	261
K.1. Modelo 3D del BMS. Vista superior. . . . .	320
K.2. Modelo 3D del BMS. Vista inferior. . . . .	321
K.3. Modelo 3D del CAN2Serial. Vista superior. . . . .	322
K.4. Modelo 3D del CAN2Serial. Vista inferior. . . . .	323
K.5. Modelo 3D del EPSolar-CAN. Vista superior. . . . .	324
K.6. Modelo 3D del EPSolar-CAN. Vista inferior. . . . .	325
K.7. Modelo 3D del Programador ISP. Vista superior. . . . .	326
K.8. Modelo 3D del Programador ISP. Vista inferior. . . . .	327

# Índice de tablas

3.1. Principales características del LTC6804. . . . .	50
4.1. Valores mínimos para el trazado de pistas de la PCB del BMS. . . . .	69
4.2. Valores mínimos para el trazado de pistas de CAN2Serial. . . . .	75
7.1. Radiación solar diaria en Los Cristianos[26]. . . . .	116
7.2. Irradiación solar diaria ( $Wh/m^2$ ) según inclinación. . . . .	116
7.3. Tabla con la relación consumo/irradiación para diferentes inclinaciones y meses. . . . .	117
7.4. Características del panel elegido. . . . .	118
8.1. Direcciones de memoria y decodificación de valores. . . . .	131
8.2. Tabla de conversión de tiempos de DCTO. . . . .	136
8.3. Tabla con valores de configuración del tipo de balanceo. . . . .	138
8.4. Tabla de conversión de tiempos de DCTO. . . . .	144
8.5. Tabla con valores de configuración del tipo de balanceo. . . . .	145
13.1. Descomposición del presupuesto general del proyecto . . . . .	211
13.2. Precio detallado de los componentes electrónicos. . . . .	212
13.3. Presupuesto elementos de fabricación de PCB. . . . .	212
13.4. Precio detallado de los componentes eléctricos de la batería. . . . .	213
13.5. Precio detallado de los elementos mecánicos. . . . .	213



# Capítulo 1

## Introducción

### 1.1. Antecedentes

El auge de los dispositivos eléctricos y electrónicos está acrecentando la necesidad de emplear sistemas de baterías eléctricas recargables de capacidades cada vez mayores, con un bajo coste y un peso y tamaño reducido. En la actualidad, existen diversos equipos autónomos que trabajan empleando conjuntos de baterías y dispositivos de generación como paneles fotovoltaicos o generadores eólicos. Tradicionalmente las baterías empleadas eran de tipo plomo-ácido, poco eficientes pero muy económicas y seguras de operar. El mercado ha evolucionado hacia baterías de Litio, sistemas con eficiencias más elevadas pero más peligrosas de operar. Por ello, estas baterías precisan de un Sistema de Gestión, denominado BMS de sus siglas en inglés de '*Battery Management System*'.

El grupo de *Bioacústica Física y multi-sensores Distribuidos (BFmSD)* del Departamento de Ingeniería Industrial de la Universidad de La Laguna realiza investigación con sistemas de monitorización acústica marina y aérea para el estudio de índices de significación eco-acústica que se quieren relacionar con el cambio climático, la contaminación acústica y la biodiversidad de distintos ambientes sonoros de Canarias. En estos proyectos surge la necesidad de emplear sistemas autónomos de gestión de la energía que permitan dotar de alimentación eléctrica suficiente a las redes de sensores inteligentes de los sistemas de monitorización.

Por otro lado, estos modelos de baterías están comenzando a emplearse en medios de transporte, siendo cada vez más comunes en vehículos y motocicletas tanto de pasajeros como de competición. En este sentido, el sistema desarrollado pretende ser compatible con estos vehículos, pudiendo ser utilizado en los proyectos educativos de *Motostudent* y *Formula Student* de la Universidad de La Laguna tanto para la gestión de las baterías de los sistemas de Baja Tensión como los de

Alta Tensión, en los prototipos que compitan en estos certámenes interuniversitarios de carácter internacional.

## 1.2. Objetivos

El objetivo principal de este proyecto es el desarrollo de un **Sistema de Gestión de Batería (BMS)**, así como el diseño de una **batería de 84 Ah** para alimentar sistemas de 12V y el cálculo de una instalación fotovoltaica para alimentar un sistema de monitorización sonora. Para la consecución de este objetivo es necesario la realización de una serie de objetivos que se relacionan a continuación:

### **Electrónica:**

- Diseñar una placa electrónica capaz de monitorizar los diferentes parámetros de las celdas de la batería, realizar un equilibrado/balanceo y protección de las celdas así como establecer una comunicación bus CAN con el resto de equipos del sistema.
- Dimensionar una instalación solar generadora para abastecer correctamente al sistema acumulador y el de monitoreo acústico.
- Realizar diferentes pruebas de carga y descarga de la batería.

### **Software:**

- Diseñar una interfaz de programación para permitir la configuración y diagnóstico del BMS.
- Diseñar las herramientas *software* para la correcta programación, diagnóstico y monitoreo del controlador y la batería.

### **Mecánica:**

- Fabricar un equipo portátil para el diagnóstico del sistema.
- Diseñar un sistema acumulador basado en celdas de Litio para abastecer al sistema autónomo de monitorización acústica.
- Diseñar una estructura mecánica capaz de proteger el sistema de baterías.

### 1.3. Cronología del desarrollo

El presente proyecto se ha desarrollado dividiendo las tareas a realizar en cinco paquetes de trabajo, que se enumeran a continuación:

1. Desarrollo y Fabricación Electrónica.
2. Diseño Software.
3. Diseño de la batería.
4. Fabricación del sistema de baterías.
5. Redacción del proyecto.

El proyecto se comenzó con el **Desarrollo Electrónico**, donde se estudió el estado del arte de los sistemas BMS empleados en la actualidad así como los diferentes circuitos integrados empleables para esta finalidad. Esta tarea duró una semana. Tras esto, se procedió al diseño electrónico del sistema BMS y la *Placa de Circuito Impreso* (PCB), tarea que ocupó 8 días. Seguidamente se diseñaron las interfaces de programación y comunicación con el BMS, así como finalmente un *módulo conversor de RS485 a CANBUS* para comunicar el cargador solar con el sistema desarrollado. Estas tres tareas duraron 11 días. Este paquete de trabajo se concluyó con el pedido de los componente electrónicos necesarios, tarea que tomó dos días adicionales. Paralelamente se comenzó con la fabricación de los circuitos impresos en el laboratorio, para unas vez recibidos todos los elementos, proceder al soldado y testeo de dichas placas.

Una vez finalizado el paquete de trabajo del **Desarrollo Electrónico**, se comenzó con el **Diseño del Software** de los microcontroladores de los diferentes módulos. Para ello, en primer lugar se diseñó el *firmware* del BMS, de forma paralela al diseño del software del configurador del BMS y posteriormente una interfaz gráfica de configuración y testeo, así como una herramienta de medición de consumo del sistema de captura de datos que se pretendía tomar como base para el cálculo. Estas tareas tuvieron una duración de 20 días la primera y 10 días cada una de las demás.

Con los sistemas electrónicos diseñados y caracterizado el consumo del sistema a alimentar; se comenzó con el diseño de la batería. Para ello, en primer lugar, se estudió durante 5 días los diferentes modelos de celdas comerciales para posteriormente realizar el pedido de las celdas seleccionadas. Una vez recibidas, se caracterizaron realizando pruebas de carga y descarga, tarea que duró 5 días. Tras esto, se dimensionó la instalación fotovoltaica y la batería necesaria para el sistema caracterizado y se diseño finalmente la batería. Estas dos tareas duraron 7 y 15 días, respectivamente.

Como tarea previa al último paquete de trabajo de **Redacción del Proyecto**, se procedió a la fabricación de la batería. Tras ello, se realizaron diferentes pruebas de carga y descarga integrando todos los sistemas electrónicos desarrollados en los primeros paquetes de trabajo. Este paquete de trabajo tuvo una duración de 29 días. Finalmente, una vez verificado todo el sistema, se procedió a la redacción de la memoria en un plazo de 24 días, siendo la fecha de finalización del proyecto el 3 de septiembre de 2021.

Para una mejor comprensión de la cronología de desarrollo de este proyecto, se puede consultar el Anexo A, donde se muestra el diagrama de Gantt en la Figura A.1 y la temporalización en la Figura A.2.

## 1.4. Distribución de la memoria

Esta memoria se distribuye en tres tomos, la *Memoria*, el *Pliego de Condiciones* y los *Anexos*.

- En el tomo de la *Memoria*, además de establecer los objetivos y cronología del proyecto en este primer capítulo, se describe el material y los métodos utilizados para el desarrollo del mismo en el Capítulo 2. En el Capítulo 3 se presenta una descripción general de los diferentes elementos del sistema. La descripción detallada de las diferentes etapas del Diseño Electrónico se relatan en el Capítulo 4 y la metodología de fabricación de los circuitos impresos en el Capítulo 5. El diseño y la fabricación de la batería se exponen en el Capítulo 6. En el Capítulo 7 se muestra el dimensionado del sistema de generación necesario para abastecer la batería y el *consumo*. En el Capítulo 8 se detalla el código desarrollado. En el siguiente capítulo se muestran las pruebas realizadas a la batería, mostrándose en el Capítulo 10 un ejemplo de configuración y diagnóstico del BMS. Tras esto, en el Capítulo 11, se muestran diferentes casos de uso de este BMS, tanto en sistemas autónomos como en vehículos. Por último, se muestran las conclusiones y líneas abiertas para finalizar la memoria.
- En el tomo de *Pliego de Condiciones* y *Presupuesto* se exponen los condicionantes que tenía el proyecto para su elaboración antes de comenzar y también el presupuesto resultante del desarrollo del proyecto así como el coste de fabricación de los prototipos.
- El último tomo de *Anexos* contiene los listados que por su tedio y longitud se han extraído del texto de la memoria para que sirvan de referencia al lector interesado.

# Introduction

## Background

The rise of electrical and electronic devices is increasing the need to use rechargeable electric battery systems with higher and higher capacities, low cost and reduced weight and size. Nowadays, there is a variety of autonomous equipment that works using sets of batteries and generation devices such as photovoltaic panels or wind generators. Traditionally, the batteries used were lead-acid batteries, which were not very efficient but very economical and safe to operate. The market has evolved towards Lithium batteries, systems with higher efficiencies but more dangerous to operate. For this reason, these batteries require a Battery Management System, called BMS (Battery Management System).

The *Physical Bioacoustics and Distributed Multi-sensors group* (BFmSD) of the *Department of Industrial Engineering* of the University of La Laguna carries out research with marine and aerial acoustic monitoring systems for the study of eco-acoustic significance indices that are related to climate change, noise pollution and biodiversity in different sound environments in the Canary Islands. In these projects, the need arises to use autonomous energy management systems to provide sufficient power supply to the intelligent sensor networks of the monitoring systems.

On the other hand, these battery models are beginning to be used in means of transport, being more and more common in both passenger and competition vehicles and motorbikes. In this sense, the developed system aims to be compatible with these vehicles, and can be used in the educational projects of *Motostudent* and *Formula Student* of the University of La Laguna both for the management of the batteries of the Low Voltage and High Voltage systems, in the prototypes that compete in these international inter-university competitions. .

## Objectives

The main objective of this project is the development of a **Battery Management System** (BMS), as well as the design of an **84Ah battery** to supply 12V systems and the calculation of a photovoltaic installation to supply a sound monitoring system. In order to achieve this objective, it is necessary to fulfil a series of objectives listed below:

### Electronics:

- Design an electronic board capable of monitoring the different parameters of the battery cells, balancing and protection of the cells as well as establishing CANBUS communication with the rest of the equipment in the system.
- Dimension a solar generating system to supply the storage system and the acoustic monitoring system correctly.
- Perform different battery charge and discharge tests.

### Software:

- Design a programming interface to allow configuration and diagnostics of the BMS.
- Design the software tools for the correct programming, diagnosis and monitoring of the controller and the battery.

### Mechanics:

- Build a portable equipment for system diagnostics.
- Design an accumulator system based on Lithium cells to supply the autonomous acoustic monitoring system.
- Design a mechanical structure capable of protecting the battery system.

## Chronology of development

This project has been developed by dividing the tasks to be carried out into five work packages, which are listed below:

1. Development and Electronic Manufacturing.

2. Software Design.
3. Battery design.
4. Manufacture of the battery system.
5. Drafting of the project.

The project began with the **Electronic Development**, where the state of the art of the BMS systems currently in use was studied, as well as the different integrated circuits that could be used for this purpose. This task lasted one week. This was followed by the **electronic design of the BMS** system and the Printed Circuit Board (PCB), a task that took 8 days. Next, the programming and communication interfaces with the BMS were designed, as well as a RS485 to CANBUS converter module to communicate the solar charger with the developed system. These three tasks took a total of 11 days. This work package was concluded by ordering the necessary electronic components, a task that took two additional days. At the same time, the manufacturing of the Printed Circuit Boards started, and once all the elements were received, the soldering and testing of the boards started.

Once the work package for the **Electronic Development** was completed, work began on the **Software Design** of the microcontrollers for the different modules. To this end, the BMS firmware was designed first, in parallel to the design of the BMS configurator software and then a graphical interface for configuration and testing, as well as a tool for measuring the consumption of the data capture system that was to be used as the basis for the calculation. These tasks took 20 days for the first task and 10 days each for the others.

With the electronic systems designed and the consumption of the system to be supplied characterised, the design of the battery began. To do this, firstly, the different models of commercial cells were studied for 5 days and then the selected cells were ordered. Once they were received, they were characterised by carrying out charge and discharge tests, a task that lasted 5 days. After this, the photovoltaic installation and the battery required for the characterised system were dimensioned and the battery was finally designed. These two tasks took 7 and 15 days, respectively.

As a previous task to the last work package of the **Project Drafting**, the battery was manufactured. This was followed by various charge and discharge tests integrating all the electronic systems developed in the first work packages. This work package lasted 29 days. Finally, once the entire system had been verified, the report was drafted within 24 days, with the project completion date being 3 September 2021.

For a better understanding of the chronology of the development of this project, please refer to the Annex [A](#), where the Gantt chart is shown in [Figure A.1](#) and the timeline in [Figure A.2](#).

## Document organization

This report is divided into three volumes, the *Report*, the *Specifications and Budget* and the *Annexes*.

- In addition to establishing the objectives and chronology of the project in this first chapter, the material and methods used for the development of the project are described in Chapter 2. A general description of the different elements of the system is presented in Chapter 3. The detailed description of the different stages of Electronic Design are described in Chapter 4 and the methodology of PCB fabrication in Chapter 5. The design and manufacture of the battery is discussed in Chapter 6. The developed code is detailed in Chapter 8. In the next chapter, the tests carried out on the battery are shown. An example of BMS configuration and diagnostics is shown in Chapter 10. Then, different use cases for this BMS, both in autonomous systems and in vehicles, are proposed in the Chapter 11. Finally, the conclusions and open lines to finalise the report are shown.
- In the Part of the *Specifications and Budget*, the conditions that the project had for its elaboration before starting it and also the budget resulting from the development of the project as well as the cost of manufacturing the prototypes are set out.
- The last part of *Annexes* contains the lists which, due to their tediousness and length, have been extracted from the text of the report in order to serve as a reference for the interested reader.

# Capítulo 2

## Material y método

### 2.1. Material

Tanto el *software* como el *hardware* empleados son de tipo *Open Source*, es decir, sus especificaciones, códigos y esquemáticos son de acceso público de forma gratuita. El término *libre* hace referencia a que este tipo de material puede ser copiado, modificado y utilizado libremente para cualquier fin.

Para la realización del proyecto se ha de trabajar sobre un sistema operativo GNU/Linux con los diferentes paquetes de *software* empleados trabajando en una misma versión en las diferentes máquinas, con el objetivo de evitar problemas de compatibilidad.

Para realizar el control de versiones de los diferentes trabajos se ha empleado *Git*. Para el diseño electrónico se emplea *KiCAD*, para el diseño mecánico se trabaja con *FreeCAD*, para el desarrollo de *software* se emplea *Python* y *C++* y para la elaboración de la documentación se utiliza  $\text{\LaTeX}$ .

A continuación se describen las diferentes herramientas *Open Source* empleadas.

#### 2.1.1. Distribución Linux

Para el desarrollo del presente proyecto es posible emplear cualquier distribución de GNU/Linux. En este caso, se ha hecho uso de la distribución *Linux Mint* [13] en su versión 20.1 con la versión del kernel 5.4.0 y con el entorno de escritorio *Cinnamon* en su versión 4.8.6.

Por otro lado, en la Raspberry Pi empleada para la herramienta de diagnóstico (que se describirá más adelante), se ha hecho uso de la distribución *Raspberry Pi OS*, basada en Debian [8] en su versión del kernel 5.10.17.

### 2.1.2. *Git*

Para poder llevar el control de las diferentes versiones del desarrollo y además poder ser compartidas con los tutores para la supervisión del trabajo realizado, se ha hecho uso de un repositorio en el que alojar los diferentes archivos de diseño y documentación del proyecto. En este caso se ha hecho uso de *Git*.

*Git* es un *software* libre diseñado para el control de versiones de todo tipo de proyectos de forma eficiente[28]. Permite un desarrollo distribuido donde los progresos realizados por los distintos miembros de un equipo se recopilan en un repositorio remoto (que puede alojarse en un servidor local o a través de internet).

### 2.1.3. *KiCAD EDA*

Para el diseño de circuitos electrónicos y placas de circuito impreso se ha hecho uso del *software* libre *KiCAD EDA*[54]. Es un programa *Open Source* para la creación de Placas de Circuito Impreso (PCB por sus siglas en inglés de *Printed Circuit Board*). Esta *suite* de gestión de diseño electrónico incorpora diferentes herramientas que trabajan con un determinado tipo de librería.

Existen dos tipos de librerías de uso en este programa: por un lado, las librerías que contienen los símbolos de los esquemáticos de los componentes se denominan *.lib* y las que contienen los *footprints* (huellas de los componentes) se denominan *.mod* o *.pretty*<sup>1</sup>. Estas librerías son de solo lectura, por ello en este proyecto se han tenido que desarrollar librerías con modificaciones de los elementos originales así como desarrollo de nuevos componentes.

#### *Eeschema*

La herramienta *Eeschema* permite el desarrollo de los esquemáticos. Estos ficheros tienen la extensión *.sch*. *Eeschema* permite generar el fichero *.net* o *netlist* que describe las conexiones existentes entre los componentes del circuito. Esta herramienta puede trabajar con tres tipos diferentes de jerarquías de esquemas:

- **Jerarquía simple:** El circuito se describe en una sola hoja de esquema, empleándose únicamente etiquetas locales para la conexión de los terminales que tengan el mismo nombre dentro de la hoja.
- **Jerarquía plana:** El esquema se describe en varias hojas de esquema al mismo nivel que deben ser conectadas entre sí con etiquetas globales (aquellas que conectan terminales de distintas hojas con el mismo nombre).

---

<sup>1</sup>La distinción entre estos dos tipos es que *.mod* permite agrupar el símbolo y la huella de un componente concreto como un conjunto denominado módulo, mientras que *.pretty* simplemente contiene huellas sin asociar a un determinado componente o módulo.

- **Jerarquía compleja:** El esquema se describe con diferentes hojas *hijas* que son instanciadas mediante bloques en una hoja *madre* o raíz, por lo que se emplea un diseño a distinto nivel, con etiquetas jerárquicas (aquellas que conectan un pin jerárquico situado en un bloque instanciado con su terminal correspondiente en la hoja de origen).

### ***CvPCB***

*CvPCB* es la herramienta de selección de huellas de *KiCAD* que permite asociar los componentes empleados en *Eeschema* con sus *footprints*. Para ello, se hace uso del *netlist* generado previamente y se modifica para asociar a cada componente una determinada huella de las disponibles en las librerías.

### ***Pcbnew***

Esta herramienta permite el diseño de la PCB. Una vez diseñado correctamente el circuito en *Eeschema* y configurado correctamente el listado de huellas y *netlist* se procede al diseño del circuito impreso.

Esta herramienta permite trabajar con hasta 32 capas de cobre y 14 capas técnicas (máscaras de soldadura, pasta de soldadura, serigrafía, etc.), así como 4 capas auxiliares de dibujo y la visualización de conexiones virtuales entre componentes, a lo que se le denomina *ratsnest*.

Además, esta herramienta permite generar los ficheros necesarios para la fabricación del circuito impreso de forma automatizada, es decir, los ficheros GERBER y EXELLON del diseño, así como la generación de los fotolitos de las diferentes capas (en formato *.svg*) para la fabricación de la placa mediante procesos químicos.

### ***GerbView* y *Bitmap2Component***

*GerbView* es la herramienta proporcionada por la *suite* para la visualización de los ficheros GERBER. Por otro lado *Bitmap2Component* permite la creación de huellas destinadas a logotipos, que pueden ser introducidas en el diseño de la PCB a partir de imágenes en formato *.png*. Esta herramienta crea un fichero *.mod* por cada una de las imágenes proporcionadas, que puede ser añadido a la placa de circuito impreso haciendo uso de la herramienta *Pcbnew*.

#### **2.1.4. *FreeCAD***

Para el diseño de las piezas mecánicas del proyecto se requiere de una herramienta de diseño paramétrico en 3D. Para este fin se ha hecho uso de *FreeCAD*

[10]. Es es una herramienta de *software* libre para el diseño asistido por ordenador en tres dimensiones. Está basada en *Open CASCADE* y programada en C++ y *Python*.

*FreeCAD* presenta un entorno de trabajo similar a otros programas de diseño como *SolidWorks*, *SolidEdge*, *Autodesk Fusion 360* o *Autodesk Inventor*. Emplea técnicas de modelado paramétrico y está provisto de una arquitectura de *software* modular, permitiendo añadir de forma sencilla funcionalidades sin tener que cambiar el núcleo del sistema.

A diferencia de los CAD tradicionales, como puede ser AutoCAD, *FreeCAD* es un CAD paramétrico, por lo que utiliza parámetros para definir sus dimensiones. En este cada elemento es tratado como un objeto definido tanto por sus coordenadas espaciales ( $x,y,z$ ) como por sus parámetros. Proporciona diferentes bancos de trabajo con los que realizar tanto diseños de piezas como añadir la tornillería necesaria y generar los planos adecuados.

### 2.1.5. *Inkscape*

Es necesario el uso de una herramienta que permita la creación o modificación de gráficos vectoriales. Para esta tarea se ha empleado el *software Inkscape*. Se trata de un editor de gráficos vectoriales gratuito y de código libre [11]. Se ha empleado para producir y editar diagramas, logotipos y los ficheros *.svg* de los folios de fabricación electrónica.

### 2.1.6. *FlatCAM*

Para la creación de los ficheros de trayectorias para máquinas CNC, denominados *G-Code*, se ha hecho uso de la herramienta *Open Source FlatCAM* [9]. Este *software* está desarrollado en *Python* y funciona tanto en GNU/Linux, como en Windows y MacOS. Con ella se han creado los ficheros de fabricación para los taladros de las PCB así como el fichero de corte del contorno.

### 2.1.7. *Prusa Slic3r*

Para la impresión de piezas en 3D es necesario un *software* que permita el laminado y el cálculo de las trayectorias de la impresora. Para ello se ha hecho uso del *software* libre *Prusa Slic3r* [16]. Esta herramienta permite convertir un modelo 3D en un conjunto de instrucciones para la impresora 3D al que se le denomina *G-Code* (como se comentó en la sección anterior), que contiene la información de los movimientos así como la temperatura y cantidad de filamento que se debe extruir en cada instante.

### 2.1.8. *Python* y C++

Para el desarrollo del *software* del proyecto se han elegido los lenguajes de programación *Python* y C++. *Python* [19] es un lenguaje de programación interpretado que soporta programación orientada a objetos, programación funcional y programación imperativa. Es un lenguaje que emplea tipos dinámicos y es multiplataforma.

Para el desarrollo del código de los microcontroladores se ha empleado C++ [7], un lenguaje con sintaxis heredada del C aunque destinado a la programación orientada a objetos. Este lenguaje de alto nivel emplea tipos de datos fijo y a diferencia de *Python* (lenguaje interpretado) es un lenguaje compilado. En este lenguaje se han hecho uso de librerías específicas proporcionadas por los fabricantes tanto del microcontrolador como de los diferentes circuitos integrados.

### 2.1.9. *QtDesigner* y *PyQT*

Para el desarrollo de las interfaces gráficas se ha empleado la herramienta *QtDesigner* [21] que permite crear interfaces gráficas en *Qt* [20] de una forma sencilla y posteriormente convertir dichas interfaces a un código *Python* haciendo uso de la biblioteca *PyQt* [18].

### 2.1.10. *Atom* y *Platformio*

Para el desarrollo del código se ha hecho uso de un editor de código llamado *Atom* [6] que permite trabajar con múltiples lenguajes así como añadir diferentes *Add-ons* como es el caso de *Platformio* [15], un entorno de desarrollo destinado a la programación de microcontroladores que permite la compilación del código, el *flasheado* del propio microcontrolador así como una conexión serial para la depuración del mismo.

### 2.1.11. $\text{\LaTeX}$ y *TeXstudio*

Para la redacción del presente proyecto se precisó de un editor de textos y fórmulas que permitiese una adecuada gestión en el sistema de control de versiones y facilitase el trabajo de maquetado y presentación. Por ello, se ha empleado  $\text{\LaTeX}$  por medio de un programa que permitiese la redacción de una forma sencilla, siendo el elegido *TeXstudio* [22].

$\text{\LaTeX}$  [12] es un sistema de preparación de documentos que destaca por una alta calidad tipográfica. Generalmente es empleado en la producción de artículos y libros científicos. Su finalidad es permitir que el autor del documento se centre en el contenido del mismo y no en su apariencia, ya que se emplean numerosas

macros que se encargan de esta función. Al igual que el resto de herramientas *software* empleadas es *Open Source*.

### 2.1.12. Máquina CNC-4030Z

Para la fabricación de las placas de circuito impreso se ha combinado la tecnología de fabricación con atacado químico con las máquinas de control numérico. La máquina de corte y fresado empleada ha sido la CNC-4030Z disponible en el aula 2.10 de la Universidad de La Laguna, que puede observarse en la Figura 2.1. Esta CNC (máquina de control numérico computerizado) tiene una resolución de  $21 \mu\text{m}$  en cada eje. Presenta un *spindle* refrigerado por una bomba de agua, garantizando una buena refrigeración en periodos largos de trabajo.



Figura 2.1: CNC-4030Z.

### 2.1.13. Horno de soldadura

Para la soldadura de los componentes SMT se ha utilizado un horno de soldadura T962 como el que se muestra en la Figura 2.2. Este dispositivo tiene una potencia de 800 W, permitiendo llegar hasta la temperatura de  $280^{\circ}\text{C}$  necesaria para el proceso de soldadura.



Figura 2.2: Horno de soldadura T962.

### 2.1.14. Impresoras 3D

Para el prototipado y fabricación de numerosas piezas se ha utilizado la tecnología de impresión 3D por deposición de material fundido (FDM). El modelo de impresora utilizada ha sido la Ender 3 Pro que se muestra en la Figura 2.3a.

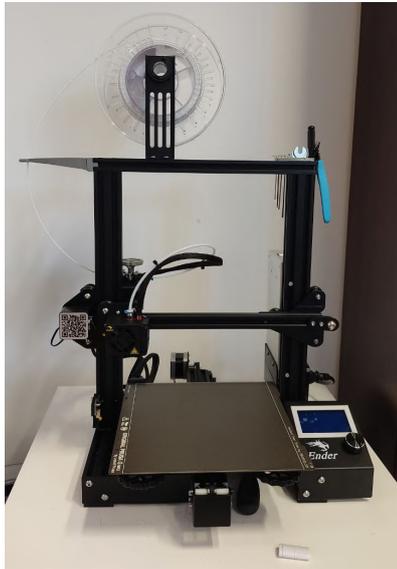
Esta impresora, que cuenta con un extrusor de tipo *Bowden*, es decir, presenta el mecanismo de desplazamiento del filamento separado del *hot-end*<sup>2</sup>, tiene una resolución máxima de movimiento de sus ejes de 60  $\mu\text{m}$ . Permite la colocación de diferentes tipos de boquillas para la extrusión, aunque se ha empleado en todas las impresiones la boquilla de 0.4 mm. Esta máquina puede imprimir en diferentes materiales como PLA, FilaFlex, ABS, PET, etc. No obstante en la fabricación de las piezas se ha empleado tanto PLA como PET-G.

### 2.1.15. Lupa electrónica

Para el análisis, la soldadura y la corrección de fallos de los circuitos impresos fabricados se ha hecho uso de una lupa electrónica. Esto ha permitido realizar diseños mucho más compactos empleando componentes SMD con encapsulados más pequeños. El modelo empleado, que puede observarse en la Figura 2.3b es una Lupa de Control Industrial TD-HD-708A disponible para su uso en el Aula 2.10 de la ESIT.

---

<sup>2</sup>El *hot-end* es el elemento de la impresora 3D que tiene como función fundir el filamento.



(a) Impresora 3D Ender 3 Pro.



(b) Lupa electrónica empleada.

Figura 2.3: Lupa e Impresora 3D empleadas.

### 2.1.16. Estación de soldadura por punto

Para la soldadura de las pletinas de níquel de la batería se ha necesitado de una máquina para soldadura por punto. Para ello, se ha hecho uso de la estación de soldadura de fabricación artesanal disponible en el aula 2.14 de la ESIT, proporcionada por el grupo *ULL Dynamics* que puede verse en la Figura 2.4. Proporciona una corriente de cortocircuito de 500 A y presenta la posibilidad de regular el tiempo de duración del pulso, permitiendo así un mayor control del acabado de la soldadura.

### 2.1.17. Carga electrónica Rigol DL3031A

Para realizar las pruebas de descarga de la batería de una forma segura y poder obtener los datos de tensión y corriente de la batería, se ha empleado la carga electrónica Rigol modelo DL3031A (Figura 2.5) disponible en el Laboratorio de Comunicaciones y Teledetección de la Universidad de La Laguna. Esta carga electrónica de 350 W puede operar con tensiones de entrada de hasta 150 V y pedir corrientes de hasta 60 A. Ofrece la posibilidad de guardar los datos en un fichero .csv en el que se almacena la tensión, corriente, potencia, energía y capacidad consumida en cada instante y acumulada, de forma que se puedan construir gráficos de descarga de la batería y estimar la capacidad extraída a la misma. Con todo ello ha sido posible desarrollar el modelo equivalente de la celda y obtener una

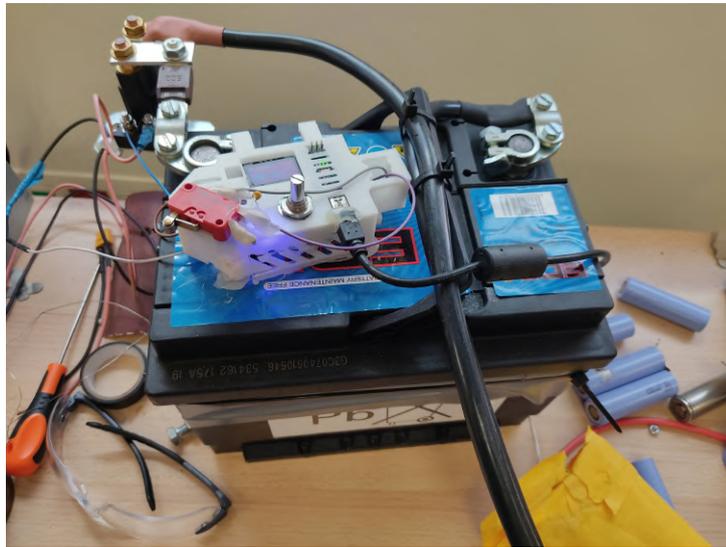


Figura 2.4: Estación de soldadura por punto empleada.

expresión para determinar el estado de carga en función de la tensión de circuito abierto de la misma.



Figura 2.5: Carga electrónica Rigol DL3031A.

## 2.2. Método

Para la elaboración de este proyecto, el método de trabajo consistió en establecer los objetivos a conseguir y determinar los materiales a emplear para lograrlos.

Una vez hecho esto, se establecieron las etapas en las que se dividiría el proyecto, donde en cada una se realizarían diferentes tareas con diferentes materiales. De este modo, fue necesario aprender a emplear correctamente los diferentes materiales, intentado que esta tarea fuese de la manera más autodidacta posible mediante la consulta de diferentes recursos. Las tareas se han realizado de forma secuencial, de este modo se debían terminar ciertas tareas clave antes de abordar las siguientes.

El uso del material antes mencionado ha permitido una total compatibilidad a la hora de revisar el trabajo realizado. Para ello, el repositorio *Git* del proyecto ha facilitado que los documentos pudieran ser compartido de una forma rápida y eficaz, permitiendo además un control de versiones óptimo.

# Capítulo 3

## Diseño e integración del sistema

Para el diseño y la fabricación de un Sistema de Monitoreo de Batería, en adelante BMS, es necesaria la integración de una serie de elementos que permitan el correcto funcionamiento de la gestión de energía del acumulador. En este capítulo, se introducen los diferentes módulos que forman el sistema, así como las interfaces de comunicación que emplean entre ellos.

### 3.1. Descripción general del sistema

En el diagrama de bloques de la Figura 3.1, pueden observarse los elementos que componen el sistema. Los elementos principales son el **nodo BMS** y la **batería**, que se encuentran alojados en una estructura mecánica que le proporciona protección frente a impactos así como de contactos eléctricos indeseados. La batería está formada por 84 celdas de Litio del modelo Samsung INR21700 [53], de 4Ah de capacidad y una tensión nominal de 3.7V, que puede oscilar según su estado de carga<sup>1</sup> (SOC) entre 2.5V totalmente descargada y 4.2V con la celda totalmente cargada. Estas celdas se han dispuesto formando una batería *4S21P*<sup>2</sup> que otorga una capacidad de 84Ah y puede operar en un rango de tensiones que abarcan desde los 9.6V hasta los 16.8V. Para la correcta operación de esta batería, se ha diseñado un BMS basado en el integrado **LTC6804** (véase Anexo E.1) de Analog Devices y controlado por un microcontrolador **ATMega 328P** (Anexo E.5). Para cargar la batería se ha decidido plantear el uso de energía solar fotovoltaica,

---

<sup>1</sup>El estado de carga (SOC por sus siglas en inglés) es un valor ente 0 y 1, o su equivalente en porcentaje, que indica la capacidad (en Ah) disponible de la batería respecto de su capacidad máxima teórica.

<sup>2</sup>Las siglas S y P hacen referencia al número de celdas en serie y en paralelo de una batería, respectivamente. Así, un 4S21P indica una batería formada por 4 grupos en serie de 21 celdas en paralelo cada uno.

por ello se ha dispuesto un cargador MPPT<sup>3</sup> y un panel fotovoltaico. El sistema a alimentar opera a una tensión de 5V por lo que debe emplearse un convertidor DC/DC que baje la tensión desde los 12V hasta los 5V. Para la correcta operación de todo el sistema, es necesario que los diferentes módulos se comuniquen entre sí. Para ello se ha hecho uso de un bus CAN, un bus muy utilizado en la industria automovilística por su robustez [37].

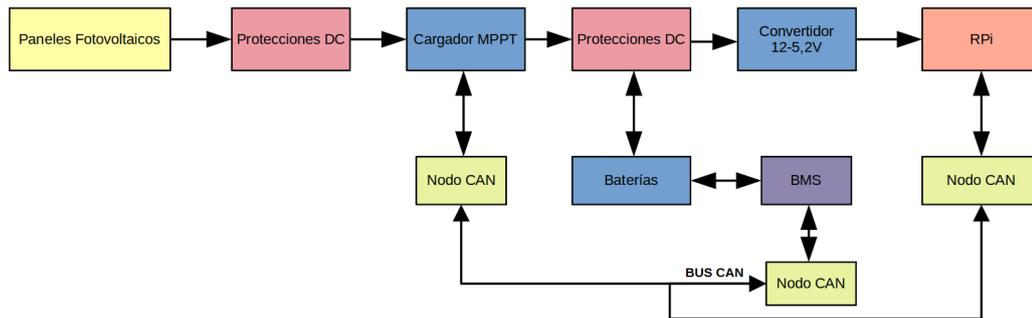


Figura 3.1: Diagrama de bloques general del sistema.

## 3.2. Descripción de los elementos del sistema

En las siguientes secciones se describen cada uno de los elementos que componen el sistema.

### 3.2.1. Sistema de Monitoreo de Baterías (BMS)

El sistema de Monitoreo de Batería, como puede verse en el Diagrama de Bloques de la Figura 3.2, está formado por un microcontrolador (ATMega 328P) (Anexo E.5) que gestiona las comunicaciones, tanto con el resto del sistema general empleando CAN bus como con el resto de periféricos de la placa electrónica. Debe comunicarse de forma local con el circuito integrado de medición de voltajes, el LTC6804 (Anexo E.1), con el circuito integrado de comunicación CAN, el MCP2515 (Anexo E.8), y con el multiplexor de 32 entradas, el ADG731 (Anexo E.4), empleado para la multiplexación de la lectura de los termistores. Para todas estas comunicaciones se ha empleado el Bus *Serial Peripheral Interface* (SPI) [36].

<sup>3</sup>Los cargadores MPPT son un tipo de cargador empleado en energía solar fotovoltaica que se caracterizan por garantizar la carga con la máxima potencia en cada instante. Presentan un algoritmo de carga que mueve constantemente el punto de operación de la curva I-V del panel solar hasta el punto en el que la potencia entregada sea máxima.

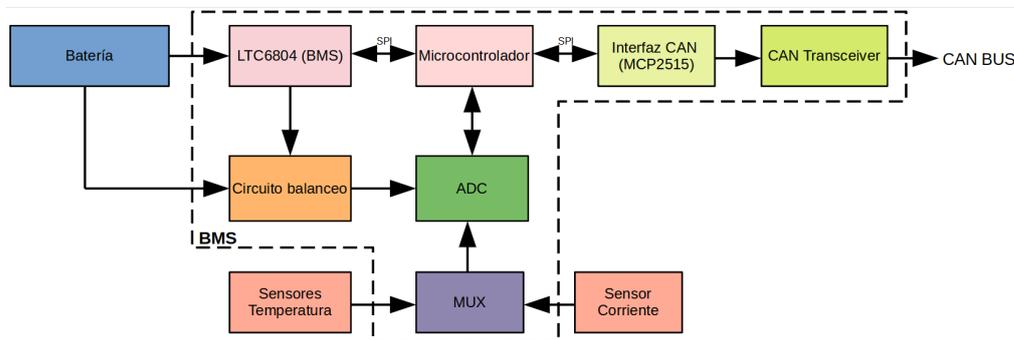


Figura 3.2: Diagrama de bloques del BMS.

El BMS diseñado permite el monitoreo de cada grupo de celdas conectadas en paralelo en una batería hasta un máximo de 12 celdas en serie, controlando la tensión del grupo y garantizando que se encuentren en el rango seguro de operación, con una diferencia mínima entre ellas. Este sistema, gestiona la conexión y desconexión de los sistemas de carga y descarga en función del estado de carga de la batería. Este parámetro es calculado internamente midiendo la corriente consumida y haciendo una estimación de la tensión de circuito abierto de la celda. Además, el sistema monitoriza la temperatura de diferentes puntos de la batería, con una capacidad máxima de medida de 32 termistores, pudiendo conocer en todo momento el perfil de temperaturas de la batería, así como la corriente de carga y descarga haciéndola operar en un rango seguro. Todos estos parámetros son totalmente configurables haciendo uso de un *software* con interfaz gráfica propiamente desarrollado para este fin que permite, mediante la interfaz de conversión serial a CAN desarrollada, enviar los comandos de configuración al BMS y monitorizar, desde un ordenador, los parámetros medidos en tiempo real.

### Microcontrolador ATmega 328P

El ATmega 328P es un microcontrolador del fabricante Atmel [4] que emplea la arquitectura AVR [5], es decir, basada en el modelo Harvard de computadores, donde las memorias de instrucciones y almacenamiento se encuentran físicamente separadas (Figura 3.3). Emplea un conjunto de 32 registros de 8 bits, siendo uno de los dispositivos más usados de la familia RISC de Atmel, y se emplea como dispositivo principal para gestionar las mediciones y las comunicaciones.

### Puertos de Entrada/Salida

Este microcontrolador cuenta con 20 pines configurables como entrada o salida según las necesidades. De estos, 6 se pueden emplear para salidas de señal

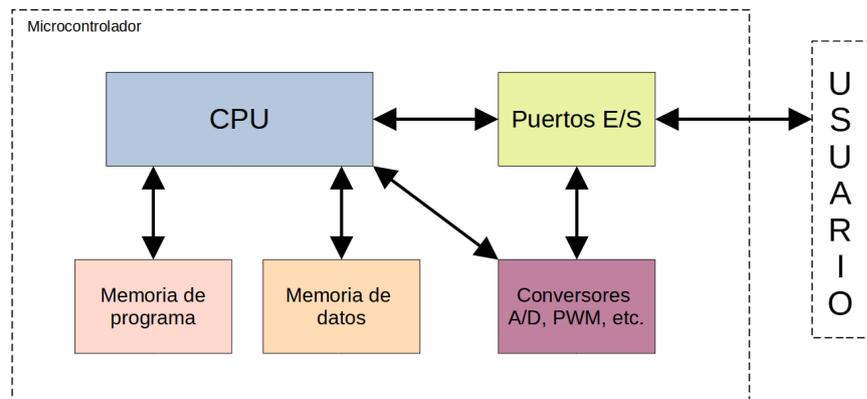


Figura 3.3: Diagrama de bloques de funcionamiento del microcontrolador.

PWM. Por otro lado, presenta 8 pines de entrada analógica multiplexadas para ser llevadas a un convertor analógico/digital (ADC) con 10 bits de resolución. Además, cuenta con un par de pines vinculados a interrupciones *hardware* del sistema, que han sido empleados para capturar la interrupción generada por el controlador CAN. Se ha hecho uso de los terminales destinados a la comunicación SPI, localizados en los pines *PB3* (MOSI), *PB4* (MISO) y *PB5* (SCK) y los vinculados a la comunicación serial (Tx y Rx), localizados en *PD0* y *PD1*, respectivamente.

### Memoria del microcontrolador

El ATmega 328P dispone de tres tipos de memoria: Flash, SRAM y EEPROM.

- **Memoria Flash:** Con capacidad de 32 KB, se emplea para almacenar el programa del microcontrolador.
- **Memoria SRAM:** En ella el microcontrolador guarda los datos que necesita para la realización de las operaciones. Se trata de una memoria dinámica y que se formatea al eliminar la tensión. En este dispositivo su capacidad es de 2 KB.
- **Memoria EEPROM:** De tipo no volátil, en ella que se almacenan aquellas variables o constantes que se pretende que permanezcan sin cambio incluso al hacer un *reset*. En ella se han almacenado los parámetros de configuración de funcionamiento del BMS (tensiones mínimas y máximas, número de celdas conectadas, número de termistores, etc.). Posee una capacidad de 1 KB.

### Integrado LTC6804

El circuito integrado LTC6804 es un integrado destinado a la monitorización de hasta 12 celdas conectadas en serie a la batería, con un error total de medida menor de 1.2 mV. Permite medir celdas de tensiones entre 0V a 5V, lo que le permite ser apto para la mayoría de modelos de celdas comerciales. La medición total de las 12 celdas toma un tiempo menor a 290  $\mu$ s. Permite la conexión de más de un integrado en serie empleando una topología en *daisy-chain*. No obstante, para la aplicación de este BMS se ha empleado un único integrado en la placa de circuito impreso controlado por un microcontrolador mediante una comunicación SPI, accediendo con este protocolo a los diferentes registros en los que se almacenan las lecturas y procediendo a su conversión.

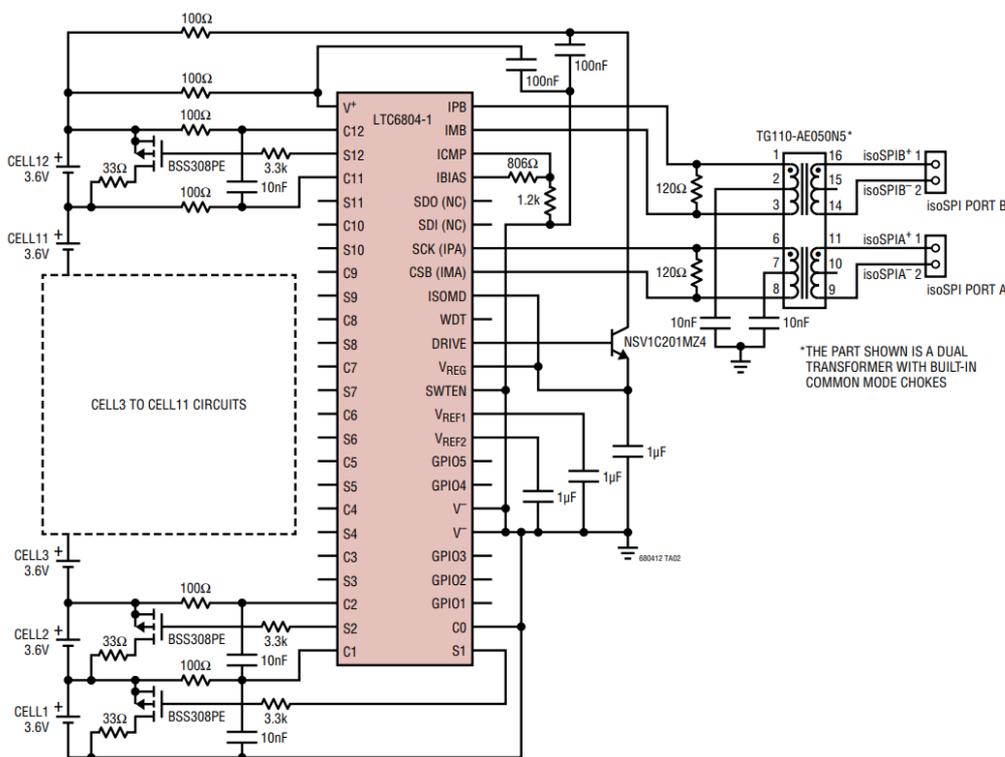


Figura 3.4: Esquema de conexión recomendado por el fabricante.[45, pág. 78]

Dispone de 13 entradas denominadas desde *C0* hasta *C12* con las que medir en los diferentes puntos de las celdas que conforman la baterías y 12 salidas (*S1* a *S12*) con las que activar los MOSFETs de control de los circuitos de balanceo de las celdas. Además dispone de 5 pines GPIO que pueden ser empleados para la medida de sensores de temperatura o corriente.

Las características principales de este circuito integrado se muestran en la Tabla 3.1.

Principales características	Valor
Tensión máxima de alimentación	75 V
Tensión en C0 respecto a tierra	-0.3 V a 0.3 V
Diferencia máxima entre V+ y C12	-5.5 V
Offset típico del ADC	0.1 mV
Error máximo de medición entre 2 celdas consecutivas	$\pm 0.8$ mV
Consumo máximo de corriente	10 mA

Tabla 3.1: Principales características del LTC6804.

### Multiplexor ADG731

El multiplexor seleccionado es un multiplexor analógico con comunicación SPI con 32 entradas. Se ha elegido un multiplexor SPI en lugar de los multiplexores tradicionales para reducir el número de pines a emplear del microcontrolador. Con este integrado, únicamente se necesita un pin adicional (el pin *SS*) mientras que para un *MUX32:1* tradicional se necesitarían 5 pines de selección.

El multiplexor seleccionado presenta una baja resistencia de conducción, siendo inferior a  $6 \Omega$  lo que lo hace ideal para la lectura de termistores de  $10 k\Omega$  como los empleados, ya que supone un error de lectura del termistor inferior al 0,1 %.

### Módulo de comunicación CAN MCP2515

Para la comunicación CAN se ha empleado un circuito integrado de la familia *Microchip*, el modelo **MCP2515**. Este controlador permite la comunicación según las especificaciones CAN en su versión 2.0B con velocidades de hasta 1Mbits/s, permitiendo el envío y recepción de tramas estándar y tramas CAN extendidas<sup>4</sup>. Dispone de 2 *buffers* de recepción de mensajes, 6 filtros de recepción y dos máscaras de 29 bits configurables. Presenta la posibilidad de generar una interrupción a la recepción de cada mensaje CAN. La comunicación del controlador CAN con el ATmega se realiza mediante bus SPI a una velocidad de 10 MHz.

Se ha seleccionado este integrado por ser ampliamente usado en la industria automovilística y por presentar unos bajos consumos de corriente, típicamente de 5mA en modo activo y 1  $\mu A$  en modo *sleep* [45]. No puede ser conectado directamente a la red bus CAN física, por lo que precisa de un *transceiver* (denominado

<sup>4</sup>La trama estándar emplea 11 bits para el campo del identificador a diferencia de la trama extendida que permite emplear 29 bits para el tramo del identificador de mensaje.

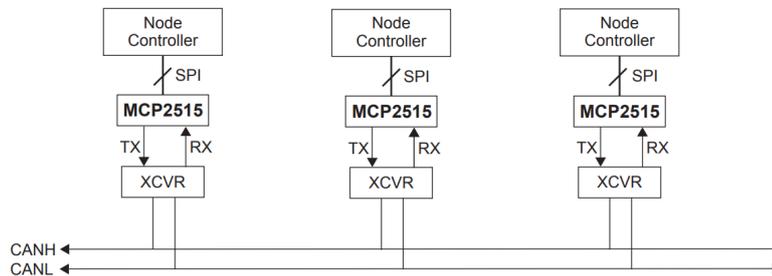


Figura 3.5: Ejemplo de implementación de un sistema con MCP2515.

*XCVR* en la Figura 3.5) que en este caso se ha empleado el **TJA1050T** para formar un esquema de implementación como el de la Figura 3.5.

### 3.2.2. Configurador de BMS *CAN2Serial*

Para poder proceder a la lectura de los mensajes del bus CAN y configurar los parámetros del BMS por medio del CAN se ha desarrollado una interfaz de conexión de CAN a Serial (Figura 3.6). Este dispositivo permite recibir por puerto serie los mensajes enviados por el BMS y permite configurar el mismo. Hace uso, nuevamente de un microcontrolador **ATMega 328P** y una interfaz CAN formada por el MCP2515 y el *transceiver* **TJA1050T**, así como un integrado **CH340** para conversión de la comunicación serie TTL a USB.



Figura 3.6: Diseño 3D de la interfaz de comunicación CAN2Serial.

### 3.2.3. Programador ICSP

Con el objetivo de poder programar los diferentes microcontroladores empleados sin necesidad de implementar una comunicación por puerto USB, a todas las

PCBs diseñadas se le ha dejado disponible un conector de programación ICSP como el que se muestra en la Figura 3.7a. Esta se conecta a un programador desarrollado (Figura 3.7b) que está formado por una conexión USB y un microcontrolador capaz de convertir las instrucciones enviadas por serial a mensajes específicos para la programación [1]. Es detectado por el ordenador como un conector serie y podrá ser programado directamente desde el editor de código *Atom*[6] con la herramienta *Platformio*[15].

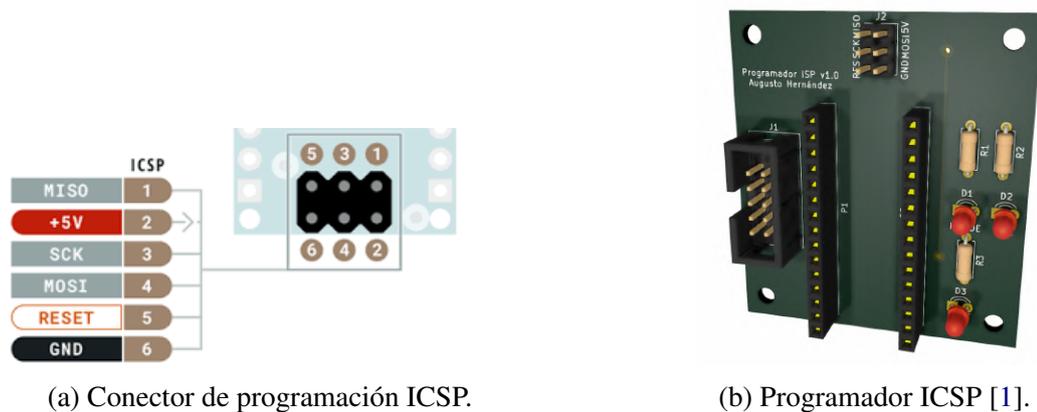


Figura 3.7: Diseño 3D del programador ICSP desarrollado.

### 3.2.4. Batería

Para alimentar el sistema de captación de datos, se ha desarrollado una **Batería de 84 Ah** (Figura 3.8) formada por celdas de ion-Litio de 4Ah con tensión nominal de 3.6V (como se mencionó anteriormente). La batería propuesta está formada por 4 paquetes de 21 celdas en paralelos conectados en serie, dotando a la batería de una tensión nominal de 14.4V apta para ser empleada con sistemas de 12V. A esta batería se le ha conectado el BMS desarrollado, haciendo uso de los pines *C0* a *C4* del mismo para la medición del voltaje de las celdas y el balanceo de las celdas.

En las baterías de litio resulta de gran importancia que la tensión de las celdas en ningún momento supere los valores límite permitidos por el fabricante, puesto que existe un alto riesgo de explosión si se incrementa por encima de la máxima o debajo de la mínima. Además, es muy importante garantizar que la carga y descarga se realicen en unas condiciones seguras. Por ello, el BMS debe monitorizar en todo momento la tensión de cada agrupación de paralelos y la corriente de entrada o salida de la batería.

Por otro lado, es sumamente importante que las diferentes celdas del *paquete* se encuentren exactamente en el mismo valor de tensión de circuito abierto y con

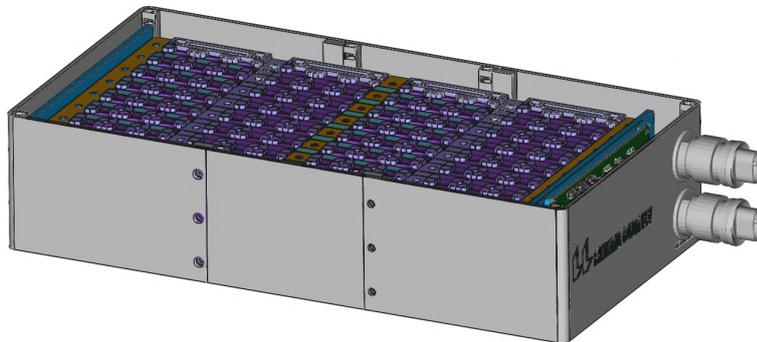


Figura 3.8: Aspecto final de la batería diseñada.

el mismo SOC. Este es uno de los mayores problemas del uso de las celdas de Litio, puesto que se debe monitorizar y corregir continuamente cualquier desviación provocada por una carga o descarga desigual de las celdas. Por ejemplo, si una celda se descarga de forma muy rápida con respecto a las demás, existe un alto riesgo de que dicha celda llegue de forma más rápida a un rango no seguro de operación, pudiendo provocar cortocircuitos o tener fugas de corrientes, lo que dañaría totalmente la batería. Para ello, se emplea el balanceo de celdas, que consiste en equilibrar el estado de todas ellas, pudiendo emplearse un balanceo pasivo o activo<sup>5</sup>. En este proyecto se ha hecho uso de un balanceo pasivo con una corriente máxima de 450 mA.

### Celdas Samsung INR21700

La batería descrita en la sección 3.2.4 se ha fabricado haciendo uso de celdas **Samsung INR21700-40T** (como puede verse en la Figura 3.9), unas celdas de Li-ion de 4000 mAh de capacidad y con una corriente máxima de descarga de 40A. Se trata de una celda de forma cilíndrica de 70 mm de alto y 21 mm de diámetro, como su referencia indica. Su tensión nominal es de 3.6V, siendo su tensión mínima (con estado de carga del 0%) de 2.6V y su tensión máxima (con SOC del 100%) de 4.2V.

Este tipo de celdas deben ser cargadas mediante un proceso de carga a CC/CV, es decir, una primera etapa de carga a **Corriente Constante**, y posteriormente, una vez superado un valor de voltaje umbral, proceder a una etapa de carga a **Tensión**

---

<sup>5</sup>En el balanceo activo se extrae corriente de una celda con un SOC mayor a las demás y haciendo uso de un convertor DC/DC *flyback* o similar que emplea esa corriente para cargar las celdas con menor SOC hasta equilibrarlas. En cambio, el balanceo pasivo consiste en descargar las celdas con mayor nivel de tensión haciendo uso de una resistencia de bajo valor en la que se disipa calor.



Figura 3.9: Celda Samsung INR21700-40T empleada.

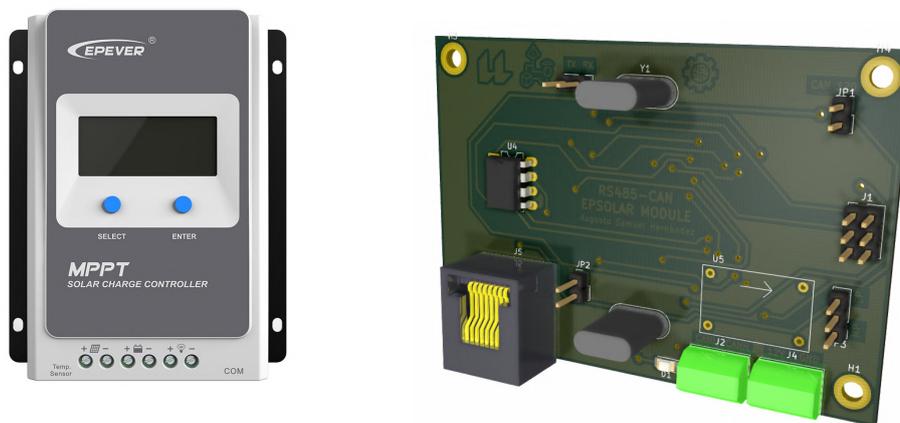
**Constante** controlando la corriente suministrada hasta alcanzar el valor de *cut-off*. Para esta celda en concreto, el fabricante establece en su *datasheet* [53] que debe ser cargada en una primera etapa a  $0,5C$ <sup>6</sup>, es decir, 2A hasta alcanzar un valor de tensión de 4.2V. A partir de ese momento, se procede a realizar una carga a Tensión Constante hasta que la corriente consumida sea  $0.05C$  (200mA). Para este proceso de carga se le estima un tiempo de 180 minutos. También existe la posibilidad de ser cargada a  $1.5C$ , es decir, a 6A hasta alcanzar el valor de 4.2V para finalizar cuando la corriente consumida descienda hasta los 100mA. Con este segundo proceso de carga, la duración es de 70 minutos, aunque la vida útil de la batería se reduce. El rango adecuado de temperaturas durante la carga es de  $0^{\circ}C$  a  $45^{\circ}C$ , siendo el rango en descarga un poco más amplio desde  $-20^{\circ}C$  hasta  $60^{\circ}C$ .

El fabricante de las celdas no proporciona una curva aproximada del SOC en función de la tensión de circuito abierto, ni un rango de valores de resistencia interna. Por ello, en el desarrollo de este proyecto se ha hecho necesaria la realización de diferentes pruebas de carga y descarga con la carga electrónica Rigol[52] para poder caracterizar correctamente la celda, identificando su resistencia interna y así obtener un modelo equivalente de la batería y con ello el estado de carga.

<sup>6</sup>C hace referencia a la capacidad de descarga nominal de la celda. En este caso 1C corresponde a 4A, siendo  $0.5C$  por tanto 2A.

### 3.2.5. Cargador solar MPPT

Para la carga de la batería diseñada se ha propuesto el uso de un cargador solar con tecnología MPPT apto para sistemas fotovoltaicos de 12V. El modelo seleccionado es el cargador **Tracer 3210AN** (Figura 3.10a) de EPSolar [30] capaz de ofrecer una potencia de 390W con una eficiencia del 98 %. Admite paneles con un punto de máxima potencia desde los 14V hasta los 72V. Es compatible con la mayoría de baterías existentes, entre ellas las de ion-Litio, al permitir configurar los valores del algoritmo de carga CC/CV. Para este caso concreto se configura el algoritmo con los valores explicados en el apartado anterior.



(a) Cargador EPSolar Tracer 3210AN. (b) Modelo 3D del convertor RS485 a CAN.

Figura 3.10: Cargador solar y sistema de comunicación.

Emplea comunicación *MODBUS* [50] mediante *RS485*, por ello se ha tenido que diseñar un sistema electrónico de conversión de *RS485* a CAN bus, al que se le ha denominado **EPSolar-CAN** (Figura 3.10b). Este sistema está compuesto por otro microcontrolador **ATMega 328P** junto con una interfaz CAN **MCP2515** y el mismo *transceiver* **TJA1050T** empleado en el resto de sistemas electrónicos.



# Capítulo 4

## Diseño electrónico

En este capítulo se muestra el diseño electrónico que se ha elaborado para los diferentes módulos que componen el sistema: El BMS, el conversor CAN2Serial, el conversor RS485 a CANBus y el programador ICSP. En las siguientes secciones se introducen cada uno de los módulos diseñados.

### 4.1. Módulo BMS

El elemento principal del diseño electrónico es el BMS. Este módulo debe ser capaz de monitorizar los parámetros básicos de funcionamiento de cada celda y controlar la carga y descarga, proporcionando además una comunicación CAN. A continuación se detallan las diferentes partes que componen el diseño de este circuito electrónico.

#### 4.1.1. Microcontrolador ATmega 328P

El elemento central de este diseño electrónico es el microcontrolador **ATmega 328P**. Este deberá realizar la comunicación con los diferentes periféricos de la placa, la estimación del estado de carga, la lectura de los sensores de temperatura y corriente y el control de las señales de habilitación de carga y descarga.

Se emplea un cuarzo de 16 MHz (denominado *Y2* en el esquema de la Figura 4.1) con 2 condensadores de 22 pF (*C9* y *C10*) y una resistencia de 1 MΩ (*R48*) para conformar el circuito oscilador del microcontrolador. La alimentación es configurable físicamente mediante *jumpers* y se puede elegir entre emplear un conversor DC/DC para obtener los 5 V a partir de la tensión de la batería o bien emplear una alimentación externa por los pines de alimentación. Esta entrada de alimentación está protegida mediante un fusible de cristal de 500 mA (denominado *F2* en el esquema de la Figura 4.1).

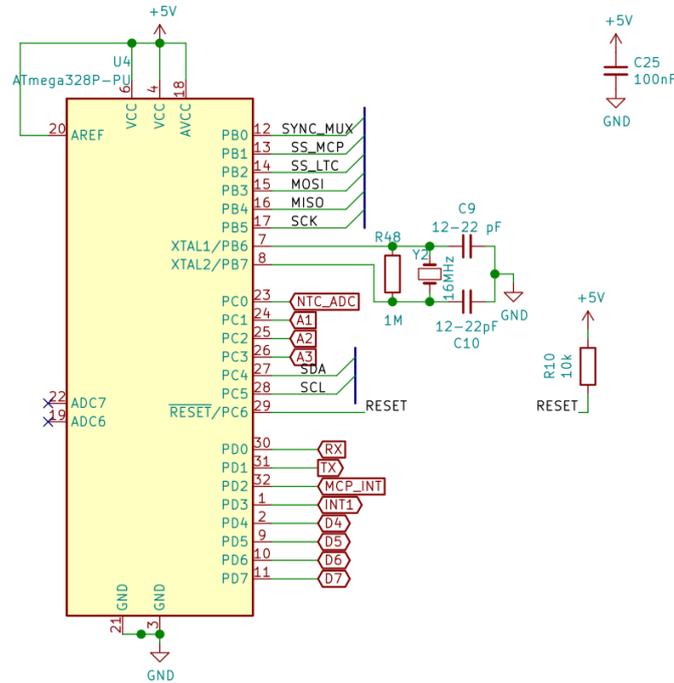


Figura 4.1: Esquema de conexiones del ATmega 328P del BMS (Anexo I).

Como se mencionó en la sección 3.2.1, se emplea la comunicación SPI como bus de comunicación con los diferentes módulos, así como interfaz de *flasheo* del *firmware*. Por este motivo, se han empleado las líneas MISO (*PB4*), MOSI (*PB3*), SCK (*PB5*) y diferentes salidas digitales como pines de selección de los diferentes módulos (*SS LTC*, *SS MCP* y *SYNC MUX*).

Por otro lado, para evitar el reseteo del microcontrolador, se ha dispuesto una resistencia de 10 k $\Omega$  entre el pin de *reset* (PC6) y la alimentación de 5V, llevando dicho punto de conexión al pin de *reset* del conector de programación ICSP (pin 5 del conector *J14*) para poder provocar el reinicio del microcontrolador durante su *flasheo*.

Además, se han llevado hacia pines externos ciertos pines digitales (*PD4*, *PD5*, *PD6* y *PD7*), así como diferentes entradas analógicas (*A1*, *A2* y *A3*) y los pines de comunicación serie Rx y Tx (*PD0* y *PD1*). Todos estos conectores pueden observarse en la Figura 4.2.

Para estabilizar la tensión de alimentación y eliminar cualquier posible ruido en ella, se han empleado dos condensadores en paralelo de 100 nF y 1  $\mu$ F, como se muestra en la Figura 4.3.

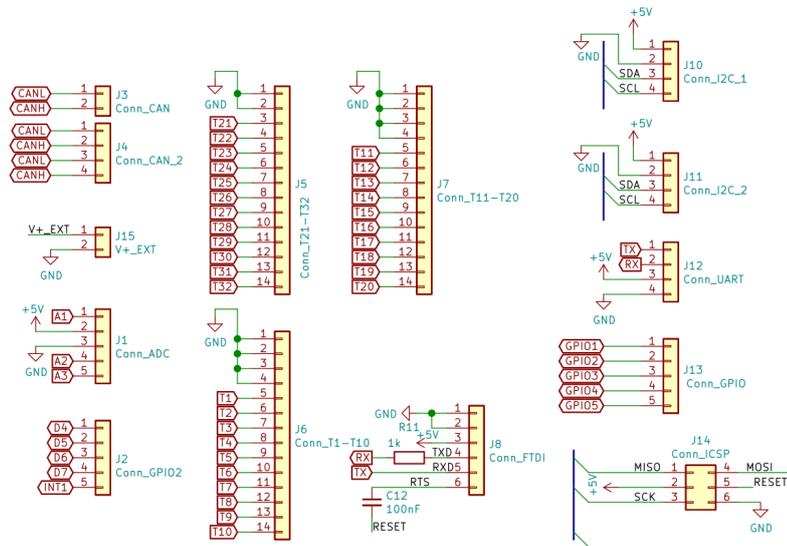


Figura 4.2: Conectores de E/S del ATmega 328P del BMS (Anexo I).

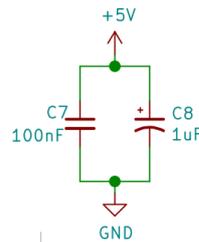


Figura 4.3: Filtrado en la alimentación del ATmega 328P del BMS (Anexo I).

#### 4.1.2. Circuito Integrado de Monitoreo LTC6804

##### Conexiones generales

Para realizar la medición y el balanceo de las celdas se hace uso del circuito integrado **LTC6804** de Analog Devices [45]. Este dispositivo, como se ha comentado anteriormente, presenta la posibilidad de emplear comunicación SPI o comunicación isoSPI<sup>1</sup>. Esto debe ser configurado eligiendo la disposición correcta de los pines *ISOMD*. Para habilitar la comunicación SPI, el pin *ISOMD* debe ser conectado a *GND* mientras que si es conectado al pin *VREG* se emplearía comunicación isoSPI. Una vez seleccionada la comunicación SPI, debe elegirse la

<sup>1</sup>isoSPI es un estándar de comunicación propio de Analog Devices, basado en el bus SPI pero que emplea únicamente dos líneas para la comunicación.

dirección mediante la conexión de los pines 45, 46, 47 y 48 a *GND* o *VREG*. En este caso, para seleccionar la dirección *0x00* se han conectado todos ellos a tierra.

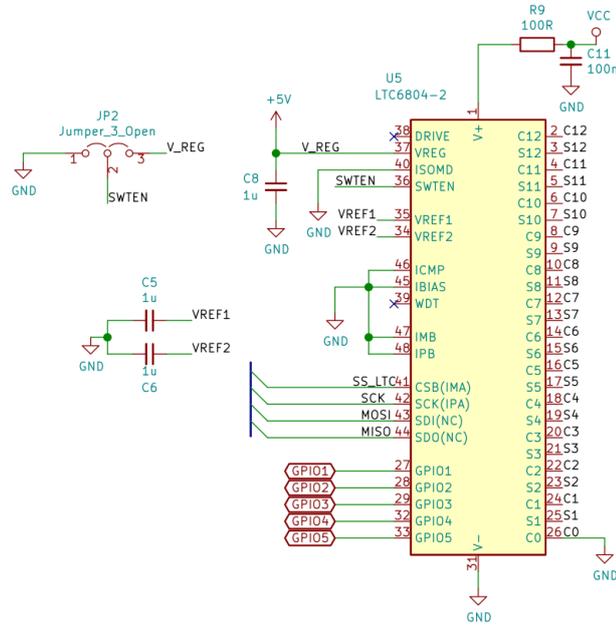


Figura 4.4: Esquema de conexión del LTC6804 (Anexo I).

La tensión de alimentación del circuito integrado se obtiene directamente desde la batería a través de un filtro paso-bajo RC con frecuencia de corte 15,9 kHz (componentes *R9* y *C11* del esquemático de la Figura 4.5). No obstante, el *datasheet* indica que la tensión de alimentación debe ser superior a 11 V; por ello, y para permitir emplear este BMS en baterías de menor tensión, se ha dotado de un *jumper* (*JP1* del esquema de la Figura 4.5) para seleccionar la alimentación de la batería o disponer de un pin de entrada de alimentación externa (localizado en el conector *J15*) para alimentar el integrado desde una fuente de alimentación de 12 V, por ejemplo.

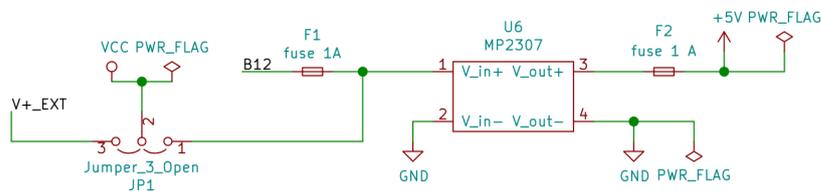


Figura 4.5: Esquema de selección de la alimentación del LTC6804 (Anexo I).

En el pin *VREG* debe disponerse una tensión de 5V para habilitar al integrado. Para ello, puede emplearse un regulador lineal de tensión a partir de transistor,

como el propuesto por Analog Devices en la Figura 4.6 o, como en este caso, emplear los 5V de salida del DC/DC empleados para alimentar el microcontrolador y demás electrónica.

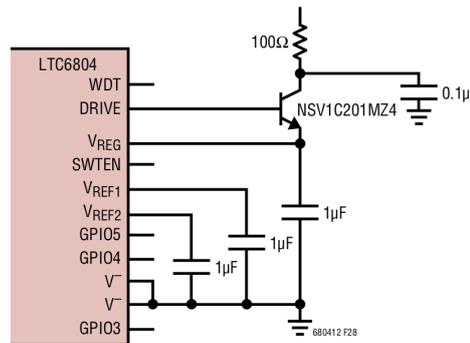


Figura 4.6: Regulador lineal de 5V propuesto por Analog Devices[45].

Por otro lado, el integrado dispone de un *timer* que regula el tiempo de *sleep* y la duración máxima de encendido de las salidas de balanceo ( $S1$  a  $S12$ ) y que puede ser configurado en dos rangos de valores diferentes. Por un lado, puede habilitarse un tiempo fijo de 2 segundos conectando el pin  $SWTEN$  a  $GND$  o bien, puede ajustarse un valor en minutos conectando el pin a  $VREG$  (5V) y configurando el valor en el registro de 4 bits  $DCTO$  según la tabla de la Figura 4.7. Por ello, nuevamente se ha dispuesto un *jumper* ( $JP2$  de la Figura 4.4) con el que elegir la conexión del  $SWTEN$ .

DCTO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Time Min	Disabled	0.5	1	2	3	4	5	10	15	20	30	40	60	75	90	120

Figura 4.7: Tabla de configuración del DCTO[45].

Los pines GPIO del LTC6804 se han dejado disponibles para el usuario, con el objetivo de poderse emplear en futuras ocasiones como expansores de entradas o salidas. Para ello, estas entradas se han llevado hacia un conector denominado  $J13$ .

### Circuito de medición y balanceo

El circuito integrado realiza la medición de la tensión en las celdas a partir de la diferencia de tensión existente entre la entrada  $C_x$  y  $C_{x-1}$ . Por ello, se disponen de 13 entradas desde  $C_0$  hasta  $C_{12}$  con las que monitorizar las 12 celdas. A estas

entradas se les ha añadido un filtro paso-bajo con frecuencia de corte 15 kHz formado por un par RC de 100 nF y 100  $\Omega$ , como el que se puede observar en la Figura 4.8. La entrada  $C_0$  está conectada al *GND* de todos los circuitos, siendo la referencia de todos los elementos el polo negativo de la batería.

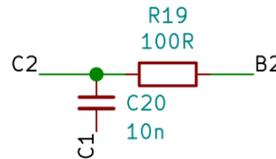


Figura 4.8: Filtro paso-bajo para medición de la tensión de las celdas (Anexo I).

En caso de emplearse una batería con menos de 12 celdas, estas deben conectarse en las entradas de numeración más baja (comenzando por  $C_0$ ), conectando todas aquellas entradas que quedan libre al punto de la batería con la tensión más alta.

Por otro lado, el LTC6804 dispone de 12 salidas de activación de balanceo que son empleadas para controlar la activación de los MOSFETs que conectan la celda con una resistencia de balanceo en la que disipar potencia y hacer bajar la tensión de la misma, produciéndose así el balanceo pasivo. Internamente estas salidas están controladas por MOSFETs de canal N [45, pág. 62] por lo que deben emplearse MOSFETs de canal P para la conexión de las resistencias de balanceo. Los transistores empleados son los **BSS308PE** [25] que conectarán o desconectarán resistencias de 10 $\Omega$ . En la Figura 4.9 se muestra el circuito de balanceo aplicado para la celda 2. Estos transistores pueden disipar una potencia máxima de 1/2W con una corriente admisible de 2A. En el caso de las celdas de ion-Litio, la tensión máxima es de 4.2V, por lo que con una resistencia de 10 $\Omega$ , la corriente máxima es de 420 mA. Con dicha corriente la potencia que disipa el transistor se calcula a partir de las Ecuaciones 4.1 y 4.2.

Como la tensión en la salida  $S_n$  corresponde al valor de tensión en el pin  $C_{n-1}$  [45, pág. 62], la tensión *puerta-fuente* es la mostrada en la Ecuación 4.1:

$$V_{GS} = V_{S_n} - V_{C_n} = V_{C_{n-1}} - V_{C_n} = -V_{cell} \quad (4.1)$$

Por tanto, siempre  $V_{GS} < V_{t_p} = -1V$ , es decir, el MOSFET está en conducción.

Como  $V_{DS} = -V_{cell} < V_{GS} - V_{t_p} = -V_{cell} - (-1)$ , el transistor está en la región de saturación, por lo que su resistencia interna se obtiene gráficamente consultando el *datasheet* (Figura 4.10).

Aproximando  $|V_{GS}|$  a 2.8V (que proporciona el peor caso con la mayor resistencia) y la corriente  $I_D$  a 420mA de corriente de balanceo, se obtiene una *Resistencia de On* de 400 m $\Omega$ , por lo que la potencia máxima disipada es la siguiente:

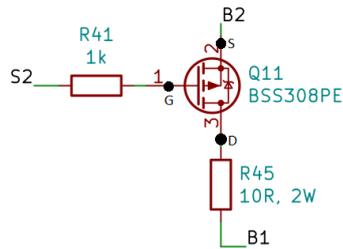


Figura 4.9: Circuito de balanceo controlado por un MOSFET de canal P (Anexo I).

$$P = 0,4\Omega \cdot (0,42\text{ A})^2 = 0,0706\text{ W} \quad (4.2)$$

Es decir, 70 mW, inferior a los 500mW de potencia máxima del MOSFET.

Por otro lado, para una resistencia de 10  $\Omega$  como las empleadas, balanceando una celda de tensión máxima 4.2 V la máxima potencia que debe disiparse es:

$$P = \frac{(4,2\text{ V})^2}{10\Omega} = 1,76\text{ W} \quad (4.3)$$

Por ello se emplean resistencias de 2W.

$R_{DS(on)} = f(I_D); T_j = 25^\circ\text{C}$

parameter:  $V_{GS}$

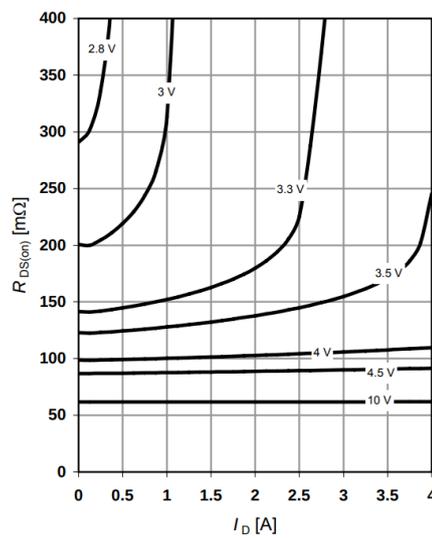


Figura 4.10: Gráfico de resistencia de conducción según  $I_D$  y  $V_{GS}$  [25, pág. 5].

### 4.1.3. Multiplexado y lectura de los termistores

Se ha decidido implementar un total de 32 entradas para sensores de temperatura. En este caso, se ha propuesto el uso de termistores NTC de valor nominal 10 kΩ, por lo que se realiza la lectura de la temperatura a partir de un divisor de tensión. El termistor se ha dispuesto conectado a tierra, siendo el circuito de sensado el que muestra la Figura 4.11.

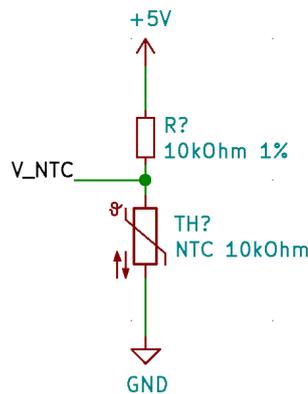


Figura 4.11: Divisor empleado para la lectura de las NTC.

Dado que el rango a medir previsto oscila entre los 0°C y los 40°C, se estudia el error de medida en el peor de los casos, es decir, una de las temperaturas extremas.

La tensión en la salida del divisor propuesto es la siguiente:

$$V_{NTC} = V_{cc} \cdot \frac{R_{NTC}}{R_{aux} + R_{NTC}} \quad (4.4)$$

Por tanto, es posible estimar el valor de la resistencia del termistor a partir de la tensión medida como se muestra en la Ecuación 4.5:

$$R_{NTC} = \frac{R_{aux}}{\frac{V_{cc}}{V_m} - 1} \quad (4.5)$$

Como la expresión de la resistencia NTC es una expresión no lineal (Ecuación 4.6):

$$R_{NTC} = R_o \cdot e^{\beta \cdot (\frac{1}{T} - \frac{1}{T_o})} \quad (4.6)$$

Despejando de la expresión la temperatura estimada, se tiene lo siguiente:

$$T = \left( \frac{\ln\left(\frac{R_{NTC}}{R_o}\right)}{\beta} + \frac{1}{T_o} \right)^{-1} \quad (4.7)$$

Sustituyendo la expresión de la medida de la resistencia NTC (Ecuación 4.5), se tiene:

$$T = \left( \frac{\ln \left( \frac{R_{aux}}{R_o} \frac{V_{cc}}{V_{NTC} - 1} \right)}{\beta} + \frac{1}{T_o} \right)^{-1} \quad (4.8)$$

Siendo la tolerancia de la resistencia auxiliar del 1%, en el peor caso la resistencia auxiliar tendrá un valor de 10.1 kΩ o 9.9 kΩ. Por tanto, para la medición de una temperatura de 40°C, se tendrá una lectura de voltaje de:

$$V_{NTC_{teorica}} = 5V \cdot \frac{5275,21\Omega}{10,000 + 5275,21\Omega} = 1,7267V \quad (4.9)$$

$$V_{NTC_{minima}} = 5V \cdot \frac{5275,21\Omega}{10,000 \cdot (1,01) + 5275,21\Omega} = 1,7155V \quad (4.10)$$

$$V_{NTC_{maxima}} = 5V \cdot \frac{5275,21\Omega}{10,000 \cdot (0,99) + 5275,21\Omega} = 1,7381V \quad (4.11)$$

Calculando a partir de la expresión 4.8, se tienen unas temperaturas medidas de 40.245°C y 39.753°C, respectivamente. Por tanto el error máximo respecto del teórico es de 0.245°C, es decir, del 0.62%.

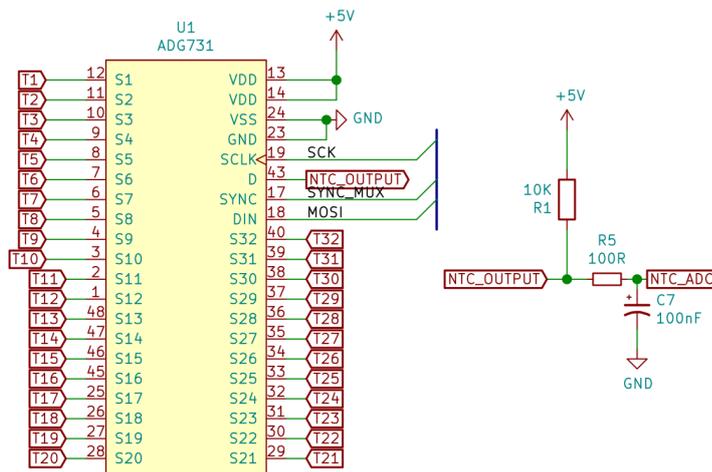


Figura 4.12: Esquema de conexión del multiplexor (Anexo I).

Para poder leer hasta 32 entradas, se ha hecho uso de un multiplexor 32 a 1 con selección de canal mediante SPI. Presenta una resistencia máxima de conducción

de  $6 \Omega$ , despreciable comparado con los valores que toma el termistor. El circuito implementado, se muestra en la Figura 4.12, donde *U1* corresponde al multiplexor **ADG731**, así como *R1* corresponde a la resistencia del divisor de tensión y los componentes *R5* y *C7* al filtro paso-bajo para eliminar el posible ruido en la lectura. La señal *NTC ADC* es llevada a la entrada *PC0* correspondiente al pin del ADC.

Se ha empleado el Conversor Analógico-Digital de 10 bits integrado en el propio microcontrolador, que para una lectura de 0-5V, ofrece una resolución de 4.88 mV, equivalente a  $0.0856^\circ\text{C}$ .

#### 4.1.4. Periféricos de comunicación CAN BUS

Para realizar la comunicación por CAN BUS se ha decidido emplear el circuito integrado **MCP2515** (integrado *U2* de la Figura 4.13), ampliamente utilizado en la industria automovilística junto con el *transceiver* **TJA1050T** (*U3*). Dicho módulo precisa de una señal de reloj para su correcta operación, por ello se ha dispuesto un cuarzo de 8 MHz (*Y1*) con dos condensadores de 27 pF (*C2* y *C3*). Para evitar el reseteo del circuito integrado, se ha dispuesto una resistencia de 10 k $\Omega$  (*R3*) desde su pin de *RESET* hacia la alimentación.

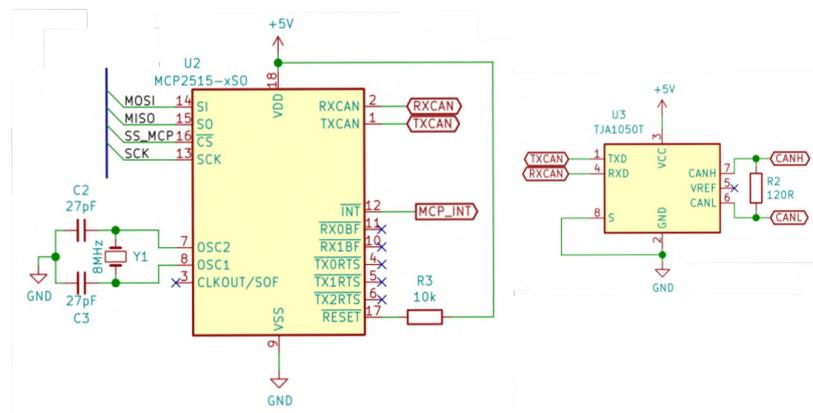


Figura 4.13: Esquema de conexión del MCP2515 y TJA1050 (Anexo I).

Las señales de salida *RXCAN* y *TXCAN* son llevadas desde el controlador CAN hasta el *transceiver*. Desde este último, se obtienen las líneas de comunicación *CANL* y *CANH*. Se ha añadido la resistencia de terminación del bus de 120

$\Omega$  (R2) entre los dos hilos del CAN, pudiendo eliminarla si el BMS no se sitúa en uno de los extremos físicos del bus.

La comunicación entre el **MCP2515** y el microcontrolador se realiza, como el resto de elementos, mediante un bus SPI. Por ello, sus pines MISO, MOSI y SCK son conectados a los pines SPI del **ATMega 328P**. Se ha decidido hacer uso de las interrupciones *hardware* en el microcontrolador. Por ello, se ha llevado la salida de interrupción de este integrado hacia la entrada INT0 (*PD2*) del microcontrolador. Cuando este periférico reciba un mensaje CAN válido tras aplicarle las máscaras y filtros establecidos, generará una interrupción en el microcontrolador, indicándole que debe procesarlo.

#### 4.1.5. Interfaces de Entrada y Salida del BMS

En cuanto a las interfaces de Entrada/Salida de este BMS, se ha hecho uso de conectores **JST** (Anexo E.15) de 14 pines para la lectura de los 13 puntos de medida de las celdas de la batería (conector *J9* del esquema de la Figura 4.14), así como 3 conectores para la medida de los termistores (conectores *J5*, *J6* y *J7*). Cada conector presenta 4 pines de GND y el resto son entradas de medida de los termistores. Además se han dispuesto conectores de tipo *pinHeader* para la lectura del sensor de corriente (*J1*), las salidas de control de los relés (*J2*) y los conectores auxiliares de comunicación y alimentación. Por último, para las conexiones CAN (*J3* y *J4*) se ha hecho uso de conectores **KF2510** (Anexo E.16), con dos pines de *CANH* y *CANL* para permitir la conexión en *daisy-chain*<sup>2</sup>.

---

<sup>2</sup>En una red CAN BUS, dado que todos los dispositivos se conectan al bus principal formado por 2 hilos. Pueden realizarse dos tipos de conexiones físicas: por un lado, emplear dos cables únicos y conectar los dispositivos mediante conectores vampiros, o emplear dos conectores en todos los elementos intermedios del bus para permitir que el cable llegue y salga del dispositivo, haciendo una conexión en *daisy-chain*.

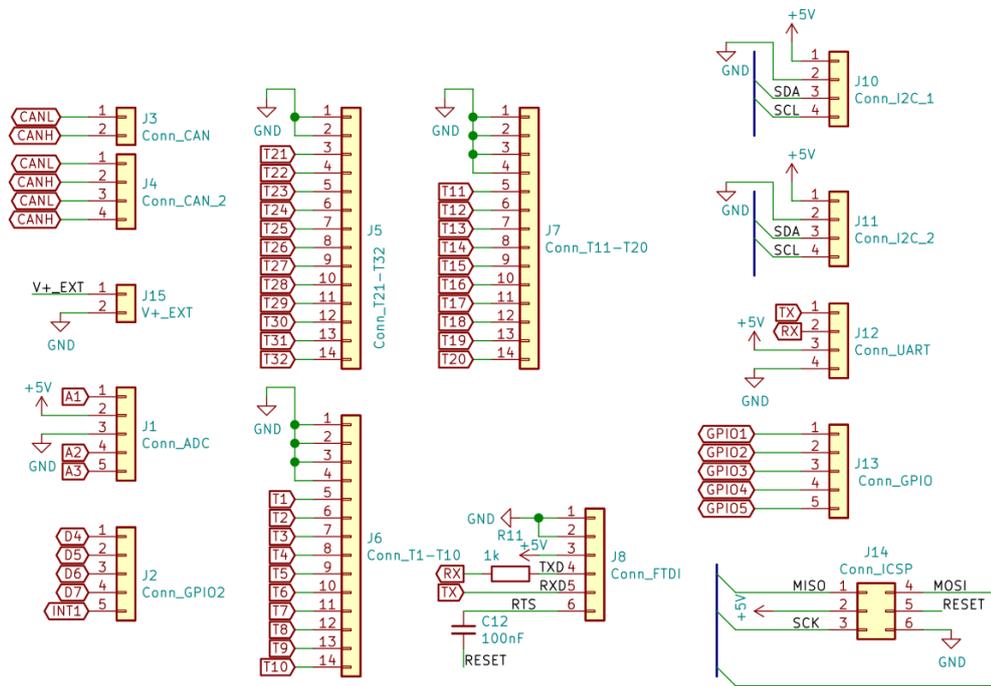


Figura 4.14: Esquema de conectores del BMS (Anexo I).

#### 4.1.6. Diseño del circuito impreso

Para el diseño del circuito impreso de la placa, se ha partido de unas restricciones dimensionales dadas por las medidas del lateral de la batería. Dado que el BMS se iba a alojar en el interior de la caja de la batería, este no podía ser superior a 150 mm de ancho y 85 mm de alto. Se partió de estas dimensiones para el diseño, obteniendo finalmente unas dimensiones de 149 mm por 83.6 mm.

Con respecto a la distribución de componentes (Figura 4.15) se trató de posicionar los integrados de forma que se minimizase el largo de las pistas del bus SPI, así como los trazados de las líneas de CAN. Por otro lado, los conectores de los termistores y la batería se han dispuesto en la parte inferior, lo más próximo entre sí, agrupando los tres conectores de las NTC de forma vertical y a continuación a la derecha el conector de la batería. El **LTC6804** se ha posicionado en el lado derecho de la placa, junto con los MOSFETs y resistencias de balanceo, de modo que en caso de producirse una excesiva temperatura en las resistencias, esta no llegue al resto de componentes del sistema.

Para el trazado de pistas se emplearon los valores mínimos que se muestran en la Tabla 4.1. Durante el trazado de las pistas se trató de aumentar al máximo posible el ancho de las pistas y vías en las zonas donde fuera posible para facilitar la fabricación y reducir la posible inducción de ruido sobre las pistas. Pueden

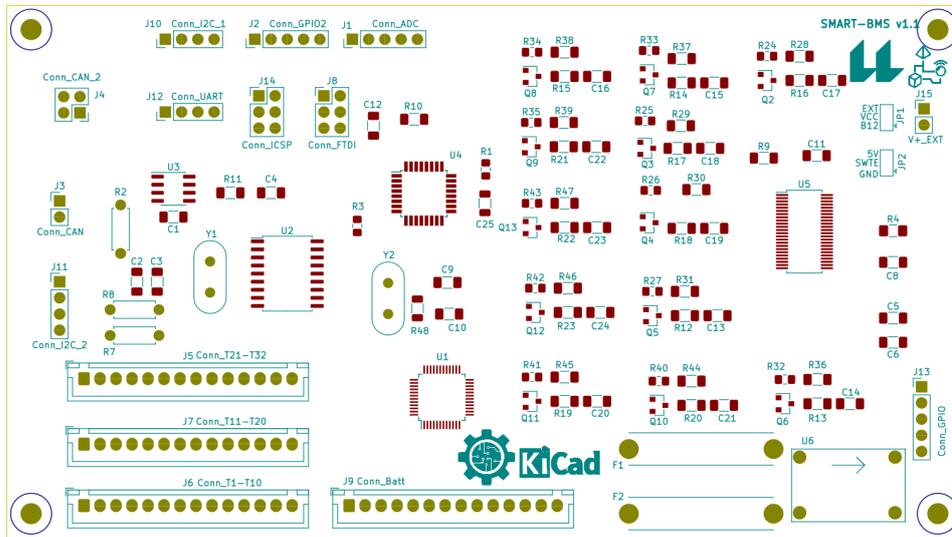
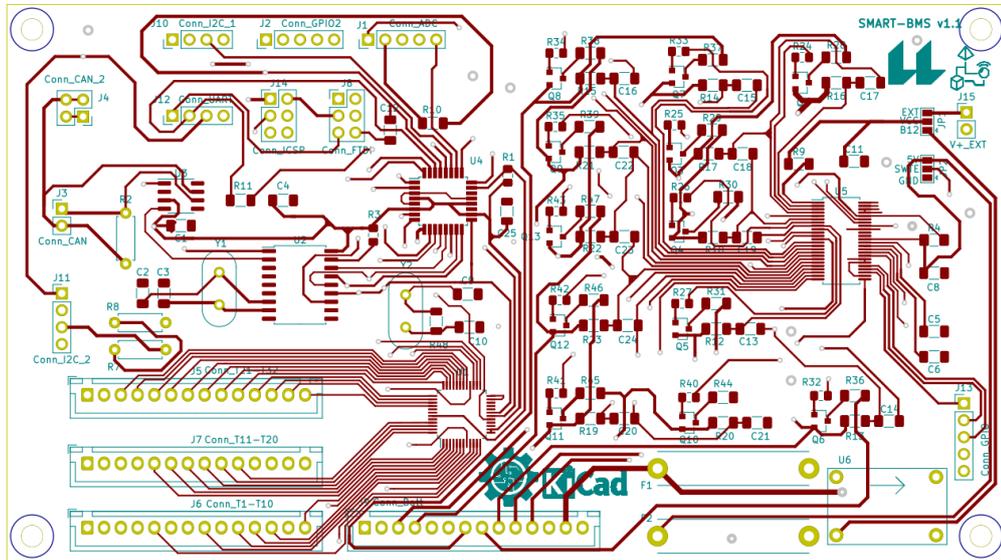


Figura 4.15: Distribución de componentes en la PCB del BMS.

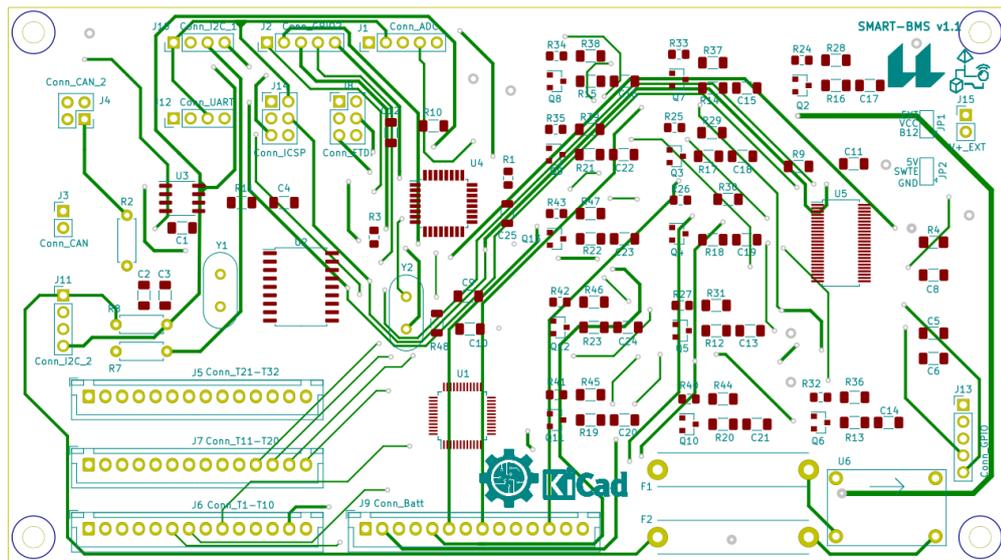
verse dos ejemplos del resultado final del ruteado de las placas en las Figuras 4.16a y 4.16b, donde las pistas rojas se encuentran en la cara superior de cobre y las verdes, en la inferior.

Margen de separación (mm)	0.2
Ancho de pista (mm)	0.35
Diámetro de vía (mm)	0.8
Diámetro de taladro vía (mm)	0.4

Tabla 4.1: Valores mínimos para el trazado de pistas de la PCB del BMS.



(a) Cara superior.



(b) Cara inferior.

Figura 4.16: Ruteado de pistas del BMS.

#### 4.1.7. Modelo 3D del circuito impreso

Con el fin de obtener una aproximación del aspecto de la placa del BMS sin fabricarse, se ha desarrollado su modelo 3D. Este modelo se ha generado a partir

del diseño en *Pcbnew* añadiendo los modelos 3D de los componentes desde las librerías de *KiCAD*. El aspecto de esta placa puede observarse en la Figura 4.17.

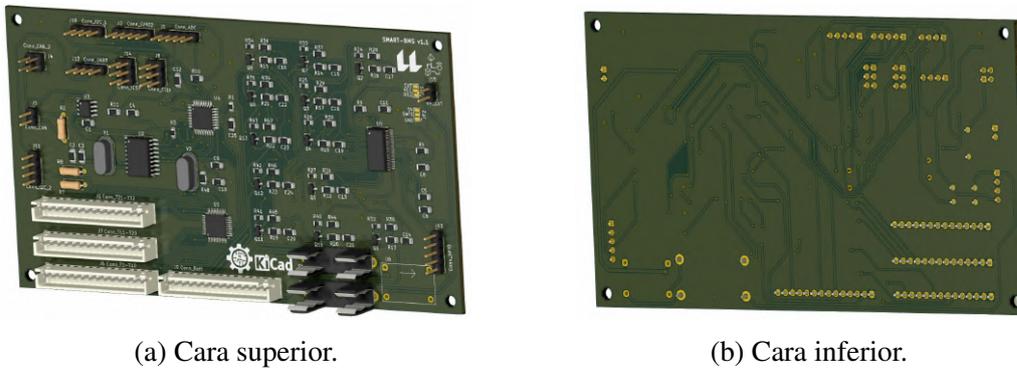


Figura 4.17: Modelos 3D del BMS.

## 4.2. Módulo conversor CAN2Serial

Para poder configurar vía CAN desde el ordenador el BMS, empleando el *software* desarrollado, debe hacerse uso de un conversor CAN a Serial. Este módulo puede conectarse al ordenador haciendo uso de un conversor serie TTL a USB o bien haciendo uso de un conversor de RS232 a USB. Por tanto, en esta placa se dispone tanto de comunicación serie TTL como RS232.

A continuación se detallan las diferentes partes que componen el circuito. Para evitar repetición, se explicarán de forma muy breve las partes de diseño comunes al módulo BMS ya explicado en la sección 4.1.

### 4.2.1. Microcontrolador ATmega 328P

El elemento central de este diseño, al igual que en el BMS, es el microcontrolador **ATmega 328P**. Este realiza la comunicación con el PC vía serial y con el módulo de comunicación CAN mediante bus SPI.

Se emplea un cuarzo de 16 MHz (denominado *Y2* en el esquema de la Figura 4.18) con 2 condensadores de 22 pF (*C5* y *C6*) y una resistencia de 1 M $\Omega$  (*R4*) para formar el circuito oscilador. La comunicación serie se realiza mediante los pines *P00* y *P01* (*Rx* y *Tx*), que también son llevados al módulo de conversión serial TTL a RS232 (el **MAX232**). Por otro lado, para su programación se ha dispuesto un conector ICSP conectado al bus SPI.

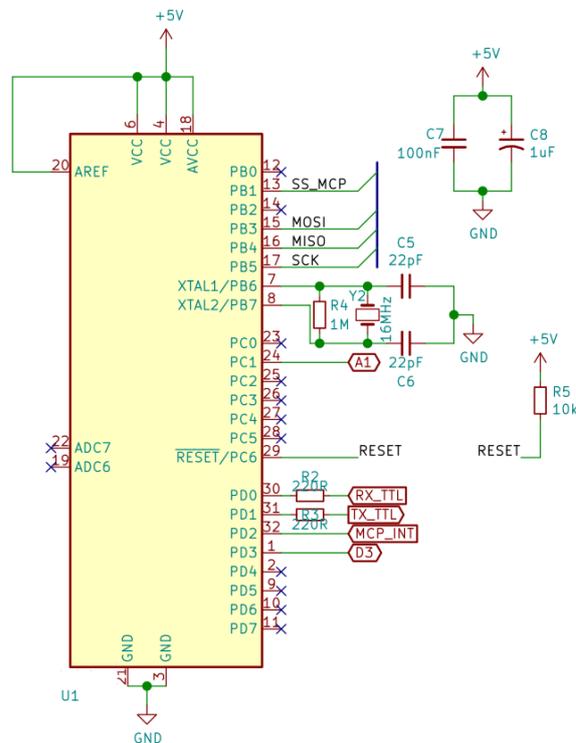


Figura 4.18: Conexión del ATmega 328P del módulo CAN2Serial (Anexo I).

#### 4.2.2. Periféricos de comunicación CAN BUS

Para realizar la comunicación CAN BUS se ha empleado nuevamente el conjunto **MCP2515** y **TJA1050T** conectado de forma similar a la explicada en la sección 4.1.4. En este caso, se ha dispuesto de un *jumper* (*JP2* del esquema de la Figura 4.19) con el que habilitar la resistencia de terminación de 120  $\Omega$ . Dado que se trata de un módulo de programación que no estará continuamente conectado al sistema general, se emplea únicamente un par de terminales de salida para el CAN.

El esquema de conexión de estos periféricos de comunicación se muestra en la Figura 4.19.

#### 4.2.3. Comunicación RS232

Para realizar la comunicación del módulo con el ordenador se ha decidido emplear comunicación serie TTL (empleando directamente los pines *Rx* y *Tx* del microcontrolador hacia un convertidor serial a USB como el **CH340**) y RS232. Para esta segunda opción es necesario emplear un convertidor de TTL a RS232,

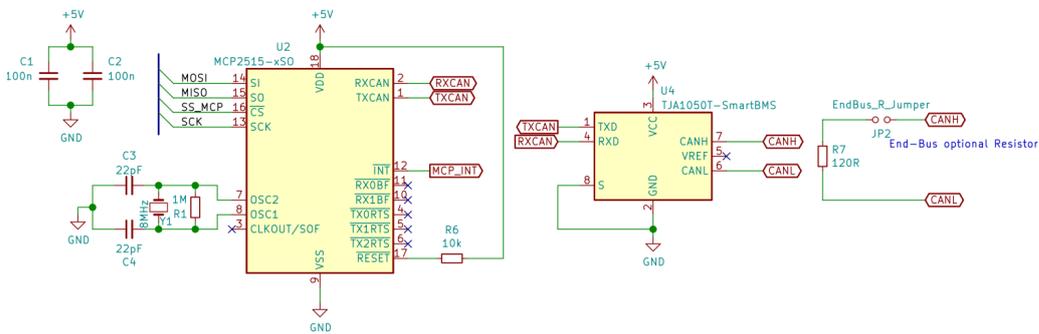


Figura 4.19: Conexión de los periféricos CAN de CAN2Serial (Anexo I).

para lo que se ha utilizado el **MAX232** del fabricante Maxim (véase el Anexo E.6). Este se ha conectado mediante los pines *Tx* y *Rx* correspondientes al canal 1 (*T1IN* conectado al *PD1* del microcontrolador y *R1OUT* conectado a *PD0* del microcontrolador, como muestra la Figura 4.20). La salida serie RS232 se obtiene en los pines *T1OUT* (que es la salida *Tx*) y *R1IN* (que es la entrada *Rx*). Para obtener los niveles de tensión mínimos de  $\pm 6V$  en las comunicaciones, deben emplearse 5 condensadores electrolíticos de  $1\mu F$  conectados según lo dispuesto en el esquema. Para poder seleccionar el uso de esta interfaz de conexión se ha dispuesto un *jumper* (*JP1*) con el que activar la alimentación del **MAX232**.

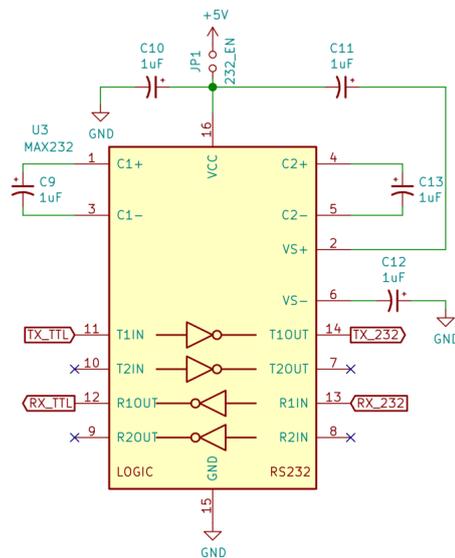


Figura 4.20: Conexión del MAX232 del módulo CAN2Serial (Anexo I).

#### 4.2.4. Interfaces de Entrada/Salida

Para la comunicación del módulo tanto por CAN como serial se han dispuesto conectores de tornillo de 5mm de paso (como se muestra en la Figura 4.21). No obstante, a estos terminales se han conectado los pines de los conectores DB9 hembra situados en la carcasa del CAN2Serial. El *pinout* de estos conectores se muestra en la Figura 4.22.

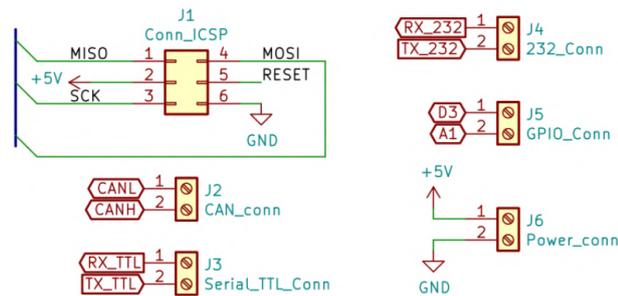


Figura 4.21: Interfaces de Entrada y Salida de CAN2Serial (Anexo I).



Figura 4.22: Pineado de los conectores DB9.

#### 4.2.5. Diseño del circuito impreso

En este diseño, a diferencia del BMS, no se partían de restricciones dimensionales. Sin embargo, dado que la intención principal de este módulo es emplearse junto con un equipo portátil, se consideró conveniente establecer unas dimensiones de 50 mm x 67 mm.

La distribución de componentes se realizó siguiendo los mismos criterios establecidos para el diseño de la PCB del BMS explicado en apartados anteriores aunque los componentes SMD, en este caso, se han dispuesto en la cara inferior

Margen de separación (mm)	0.3
Ancho de pista (mm)	0.35
Diámetro de vía(mm)	1.6
Diámetro de taladro vía (mm)	0.8

Tabla 4.2: Valores mínimos para el trazado de pistas de CAN2Serial.

con el objetivo de minimizar el número de vías. En la Tabla 4.2 se muestran los valores mínimos del trazado de pistas de esta placa, mientras que la Figura 4.23 muestra el trazado y disposición de los componentes.

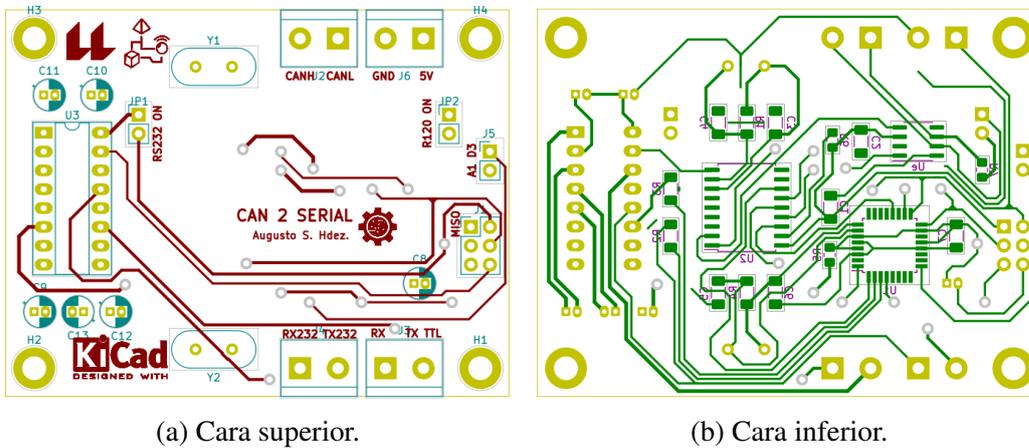


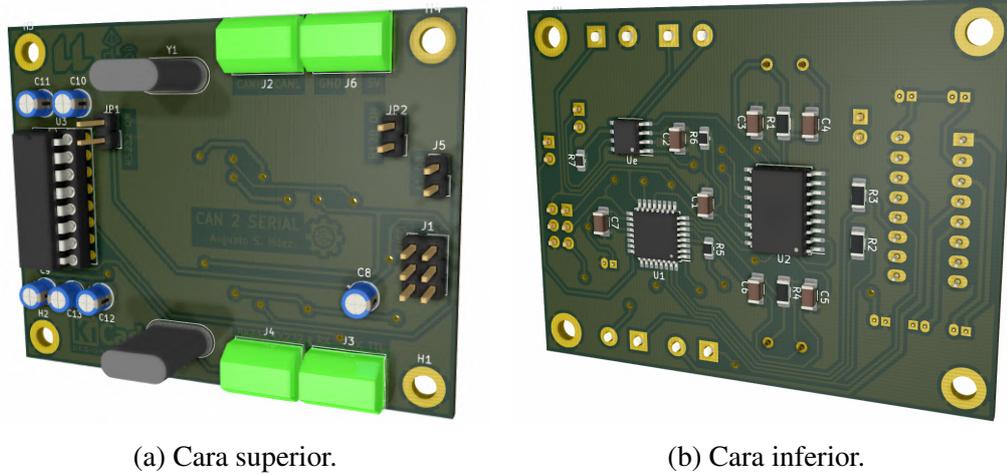
Figura 4.23: Ruteado de pistas del CAN2Serial.

#### 4.2.6. Modelo 3D del circuito impreso

Para obtener una aproximación de la placa sin fabricarse y poder emplearla para el diseño de la carcasa que se ha impreso en 3D, se ha generado el modelo 3D siguiendo la misma metodología explicada en el diseño del BMS. A continuación, en la Figura 4.24 se muestran los modelos 3D de la PCB.

### 4.3. Módulo convertor RS485 EPSolar-CAN

El cargador solar propuesto, tal y como se comentó en la sección 3.2.5, emplea comunicación MODBUS mediante RS485. Dado que se deseaba que todo el sistema pudiera comunicarse entre sí y el BMS pudiese enviar mensajes CAN



(a) Cara superior.

(b) Cara inferior.

Figura 4.24: Modelos 3D del CAN2Serial.

para controlar el encendido tanto de la carga de la batería como la alimentación de los circuitos de consumo (conectados físicamente a los relés internos del cargador), fue necesario incluir un módulo puente que realizara la conversión de los mensajes CAN a mensajes MODBUS. Para ello, nuevamente se hizo uso de un microcontrolador **ATMega 328P** con una interfaz CAN formada por el **MCP2515** y **TJA1050T** conectadas de igual forma que el módulo CAN2Serial. No obstante, en este circuito se ha añadido el circuito integrado **MAX485** de Maxim para realizar la comunicación RS485 (Anexo E.7).

### 4.3.1. Comunicación RS485

Para la comunicación MODBUS se ha empleado el integrado **MAX485** (*U4* de la Figura 4.25) conectando los 2 pines de comunicación (*DI* y *RO*) a *PD5* y *PD6*, donde se emplea una UART definida por *software* en el microcontrolador, permitiendo así mantener libre la UART original para emplearla en una posible comunicación serial con un ordenador, y conectando los pines ‘*Receiver Enable*’ y ‘*Driver Enable*’ a dos salidas digitales: *PD3* y *PD4*, respectivamente. Estas 4 señales se han conectado con sus respectivas resistencias de *pull-up* de 10 k $\Omega$  (*R7*, *R8*, *R9* y *R11*).

A la salida del bus RS485 se han dispuesto las resistencias de *pull-up* y *pull-down* de 20 k $\Omega$  (*R14* y *R15*) así como la resistencia de terminación de 120  $\Omega$  (*R13*).

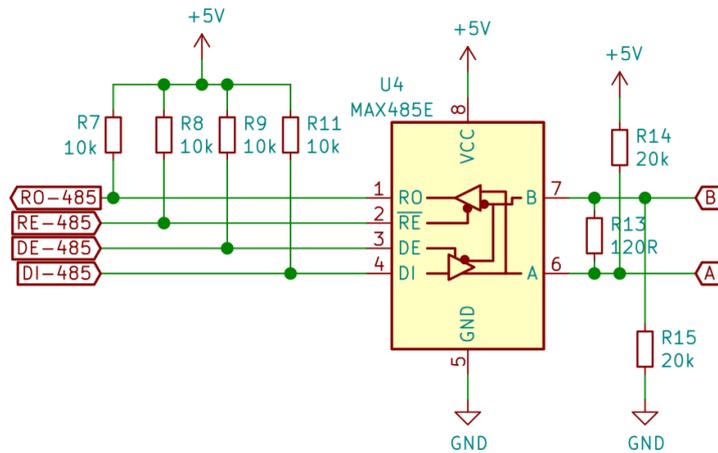
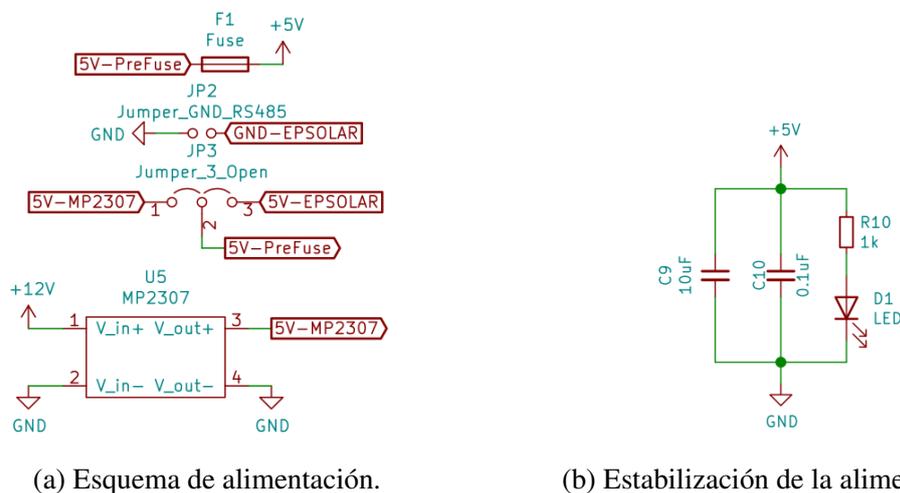


Figura 4.25: Conexión del MAX485 en el módulo EPSolar-CAN (Anexo I).

### 4.3.2. Alimentación del módulo

Para la alimentación del EPSolar-CAN se ha dispuesto un *jumper* (JP3 de la Figura 4.26) para elegir entre la entrada de 12V de la batería o la alimentación de 5V directa proporcionada por el cable de comunicaciones con terminales RJ45 del cargador solar, así como otro *jumper* (JP2) para unir las tierras de ambos sistemas.



(a) Esquema de alimentación.

(b) Estabilización de la alimentación.

Figura 4.26: Circuitos de alimentación del convertor EPSolar-CAN (Anexo I).

En caso de elegirse la primera opción, la tensión de 12V será bajada hasta los 5V empleando el mismo convertidor DCDC que emplea el BMS (el MP2307).

La salida de ambas alimentaciones es llevada a un fusible de 500mA (*F1* en el esquema) antes de alimentar todo el sistema.

Por otro lado, esta alimentación de 5V es estabilizada y filtrada haciendo uso de 2 condensadores de  $10\mu\text{F}$  y  $100\text{nF}$  (*C9* y *C10*) en paralelo. En añadido, se ha dispuesto un led (*DI*) con el que verificar que el sistema se encuentra correctamente alimentado

### 4.3.3. Interfaces de Entrada/Salida

Para la comunicación RS485 se ha hecho uso de un conector *RJ45* con el conexiado según lo mostrado en la Figura 4.27 para proporcionar la misma interfaz física que el cargador solar. Para las conexiones de la alimentación (*J4*) y comunicación CAN (*J2*) se ha hecho uso de *terminal blocks*. Únicamente se han empleado *pinHeaders* en la interfaz ICSP de programación y la conexión serial de depuración.

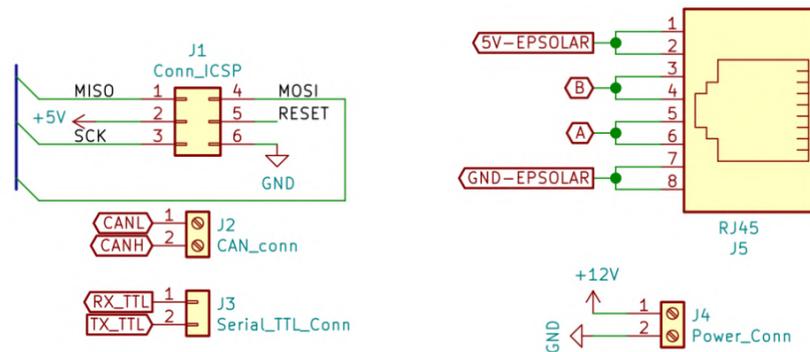
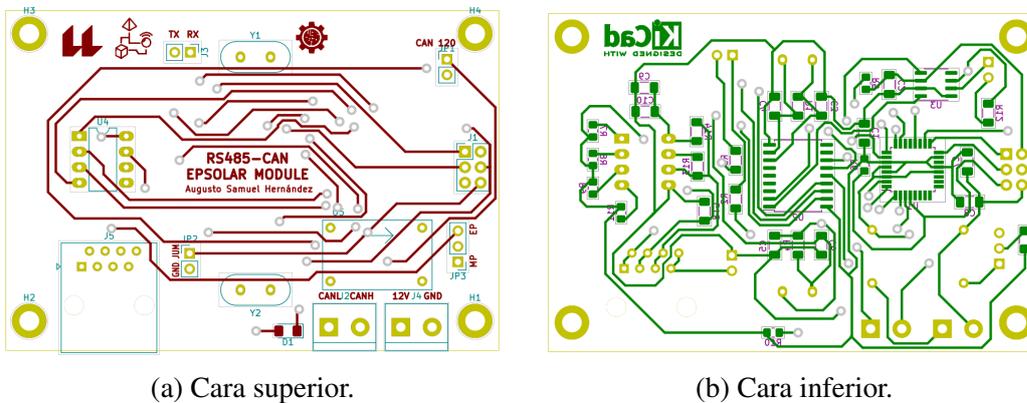


Figura 4.27: Interfaces E/S del módulo EPSolar-CAN (Anexo I).

### 4.3.4. Diseño del circuito impreso

Al igual que sucedía en la placa de circuito impreso del CAN2Serial, no se dispone de restricciones dimensionales. Sin embargo, se ha definido unas dimensiones de 80 mm de largo y 55 mm de alto.

Los componentes SMD se han situado en la cara inferior, dejando la cara superior para los conectores y los componentes THD. En la Figura 4.28 puede observarse el trazado y disposición de componentes. En cuanto a los valores mínimos para el trazado de pistas, son los mismos que los establecidos en la Tabla 4.2.



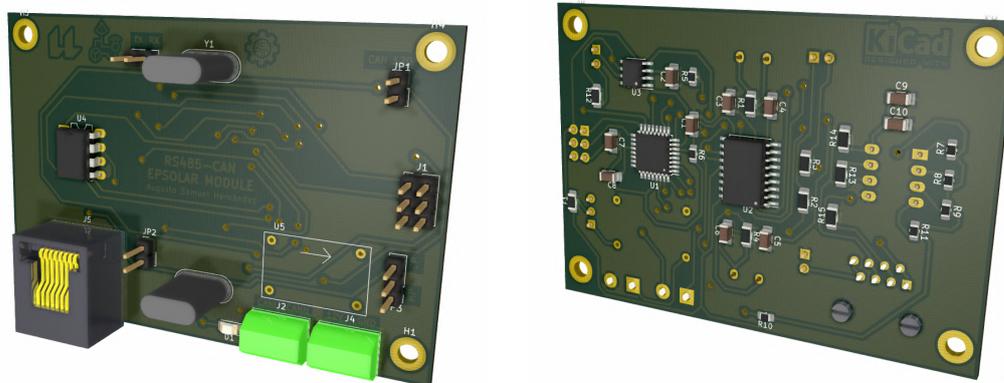
(a) Cara superior.

(b) Cara inferior.

Figura 4.28: Ruteado de pistas del módulo EPSolar-CAN.

### 4.3.5. Modelo 3D del circuito impreso

Nuevamente se ha generado el modelo 3D a partir de las librerías de componentes empleando el módulo *Pcbnew* de *KiCAD*. Dicho modelo se muestra en la Figura 4.29.



(a) Cara superior.

(b) Cara inferior.

Figura 4.29: Modelos 3D del módulo EPSolar-CAN.

## 4.4. Programador ICSP

La última de las placas diseñadas es un módulo programador ICSP. Este módulo, como se comentó en apartados anteriores, tiene como objetivo programar los

microcontroladores sin necesidad de extraerlos de la PCB, empleando el protocolo serie JTAG [1]. Por tanto, convierte los mensajes enviados por el puerto serie a mensajes específicos para el *flasheo* del microcontrolador AVR.

Para esta función se ha decidido emplear un **Arduino Nano** ya que cuenta con una conexión USB incorporada y puede ser reprogramado mediante dicha interfaz. Para emplearlo se ha diseñado una PCB con un zócalo de conexión de forma que puedan extraerse cómodamente las señales que deben llevarse al AT-Mega 328P así como unos leds para verificar el progreso durante el *flasheo* del *firmware*.

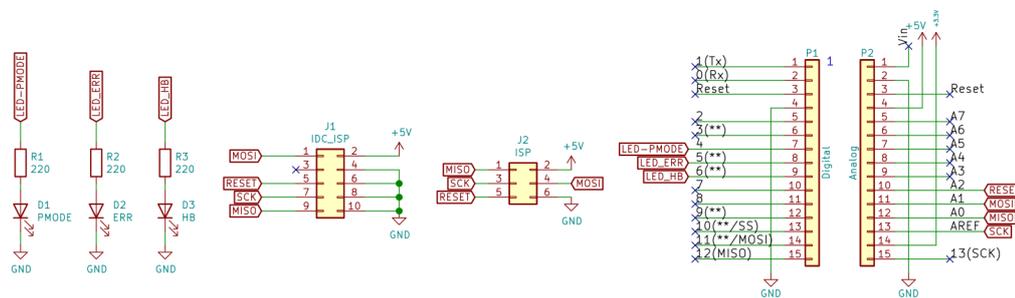


Figura 4.30: Esquema de conexión del programador ICSP (Anexo I).

Como se puede observar en la Figura 4.30, la PCB dispone de un conector de 6 pines ICSP igual al empleado en todos los circuitos desarrollados y 3 leds con los que se indica la presencia de un error durante el *flasheo* (color rojo), el *heart-beat* (color azul) y la *programación en curso* (color amarillo). La alimentación se obtiene desde el propio puerto USB.

A continuación, en la Figura 4.31 se muestra el aspecto de las pistas ruteadas tanto en la cara superior como la inferior. En cuanto a los valores mínimos para el trazado de pistas, son los establecidos en la Tabla 4.2.

#### 4.4.1. Modelo 3D del circuito impreso

Nuevamente, se ha generado el modelo 3D a partir de las librerías de componentes empleando el módulo *Pcbnew* de *KiCAD*. Dicho modelo se muestra en la Figura 4.32.

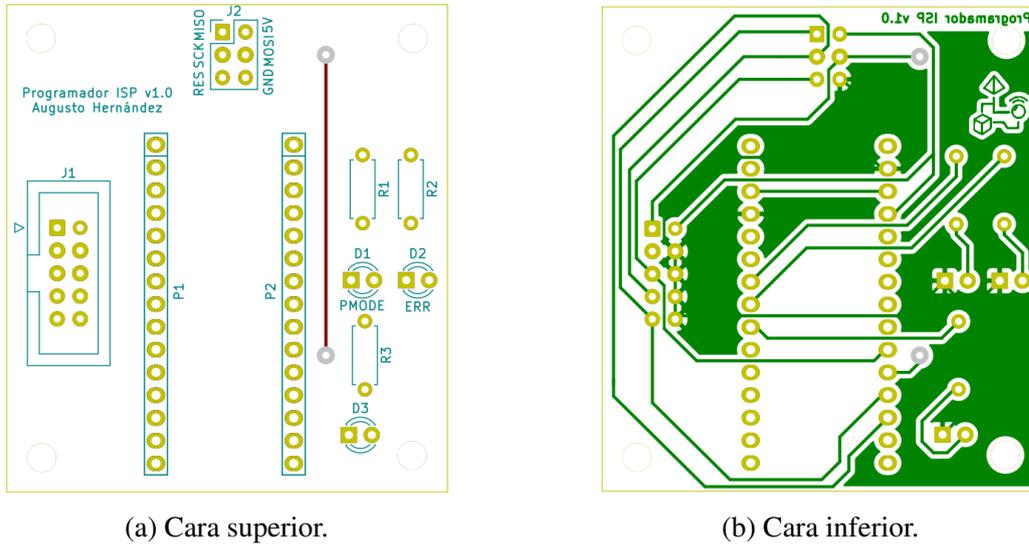


Figura 4.31: Ruteado de pistas del programador ICSP.

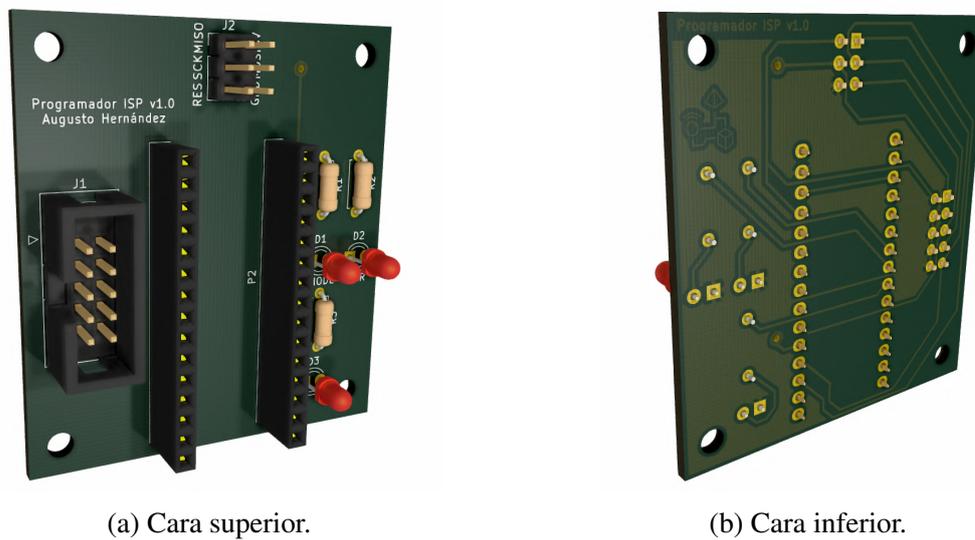


Figura 4.32: Modelos 3D del Programador ICSP.



# Capítulo 5

## Fabricación de PCBs

En el presente capítulo se introducen los aspectos y pasos a tener en cuenta para la fabricación de las placas de circuito impreso de este proyecto. Las placas a fabricar son el módulo **CAN2Serial**, el módulo **EPSolar-CAN** y el **programador ICSP**. El módulo BMS, tras realizarse un prototipo rápido, se ha encargado su fabricación al fabricante *JLCPCB* [39]. No obstante, la revisión y el soldado de los componentes se ha realizado en el laboratorio siguiendo el procedimiento que se describe en el presente capítulo.

### 5.1. Características físicas de las placas

En el capítulo anterior se describieron las características técnicas que deben tener las placas a la hora de ser diseñadas. Sin embargo, también es necesario definir las características físicas que deben tener para poder ser fabricadas. Por ello, además de la información que aportan los ficheros GERBER, se deben tener en cuenta el tipo de placas de cobre con las que se pretende realizar la fabricación y sus especificaciones:

- **Espesor de la PCB:** El espesor del sustrato de la placa debe ser de 1.6 mm.
- **El grosor de las láminas de cobre conductoras** debe ser igual o superior a 35  $\mu\text{m}$ .
- **Color de la máscara:** El color de la máscara de soldadura ha de ser verde en caso de aplicarse. En su defecto se puede aplicar laca transparente para proteger la placa.

## 5.2. Metodología de fabricación de PCBs

En esta sección se explicará la metodología seguida para la fabricación de las placas de circuito impreso del presente proyecto. Se ha empleado un método combinado de fabricación con CNC y atacado químico. Las fases de taladrado y corte se han realizado mediante CNC, mientras que el trazado de pistas se ha realizado con el atacado químico.

### 5.2.1. Fabricación de placas de circuito impreso con CNC

Para el taladrado y corte del contorno de la PCB se ha hecho uso de una máquina de control numérico (la CNC 3040Z indicada en el Capítulo 2). Para ello es necesario generar algunos ficheros de diseño durante la fase de *diseño del circuito impreso*. Estos ficheros son los siguientes:

- **Ficheros GERBER** (con extensión *.gbr*): A la hora de trazar el diseño se deben emplear los ficheros GERBER de las capas que se desean fabricar por medio de la CNC. En caso de realizarse el trazado de pistas con dicho método, se hace uso de las capas superior e inferior de cobre, así como la capa de contorno (*edge cuts*). En este caso concreto, como el trazado se realiza con atacado químico, únicamente se hace uso de la capa de contorno.
- **Fichero de taladros** (con extensión *.drl*): Como los taladros se han realizado mediante la CNC, se debe generar dicho fichero.
- **Fichero de posiciones de huellas** (con extensión *.pos*): Si se dispusiera de una máquina capaz de posicionar de forma automática los componentes, se puede generar un fichero que contenga la posición de cada huella para que la máquina coloque de forma automatizada dichos componentes.

Una vez generados los ficheros anteriores, se procede a fabricar los archivos de trayectoria de la máquina, es decir, los *G-Code*. Para generar estos ficheros, se deben conocer las características tanto de la máquina como de la herramienta a emplear. Las operaciones básicas que se pueden realizar con la CNC son las siguientes:

- *Drilling*: Operación de taladrado. En esta operación se realizan todos los taladros de la PCB.
- *Milling*: Operación de eliminación de cobre de la PCB. En esta etapa se generan las pistas del circuito impreso.

- *Cutting*: Operación de corte del borde de placa. En esta operación se ejecuta el recorte del circuito impreso. Típicamente, es la última operación a realizar.

Para generar los ficheros *G-Code* necesarios para la máquina, se ha hecho uso del *software* FlatCAM. En este programa se especifica el tamaño de la herramienta, así como la operación a realizar con sus ajustes particulares y se genera el fichero de trayectorias que posteriormente es llevado a la CNC.

Una vez finalizada la generación de dichos *G-Code*, se cargan en la máquina y se procede a colocar en la base de la CNC una placa de cobre del tamaño adecuado, correctamente sujeta. Se sitúa la herramienta adecuada y se ejecutan los *G-Code*, comenzando por los taladros y finalizando con las operaciones de corte. Después de realizar todas las operaciones con la CNC, se procede al revelado de las pistas del circuito y el posterior atacado químico.

### 5.2.2. Fabricación de placas de circuito impreso con métodos químicos

Para la producción de las pistas del circuito impreso con procesos químicos es necesaria la generación de ciertos ficheros durante la *fase de diseño de la PCB*, como ocurría con las etapas de fabricación con máquinas CNC. En este caso se deben generar las capas de cobre superior e inferior (en formato *.svg*). En el caso de la capa inferior, debe generarse en trazado invertido, porque de lo contrario al emplear tanto la técnica de *isolado* como la técnica de *transferencia de toner*, esta quedaría del revés. Una vez generados los ficheros, se emplea *Inkscape* para imprimirlos asegurándose que no haya cambios de escala en las dimensiones del diseño.

En este caso concreto, se ha hecho uso de la técnica de *transferencia de tóner*. Para ello, los fotolitos se han impreso en un papel con película de cera de tipo *transfer* haciendo uso de una impresora láser. Una vez impresos y recortados estos fotolitos, se posicionan en la PCB, verificando el correcto alineamiento con la placa de cobre. Para ello, se hace uso de la marca del contorno. Posteriormente, se coloca la placa de cobre entre los fotolitos y se aplica calor empleando una plastificadora de rodillos. Tras pasar 8 minutos de aplicar calor, se retira el papel encerado y se comprueba la correcta transferencia de la tinta. En caso de que haya pistas mal transferidas, se corrigen los trazados haciendo uso de un *rotulador permanente*.

El siguiente paso a realizar es el ataque químico de la placa. Este procedimiento se lleva a cabo para eliminar el cobre que no ha quedado cubierto por la transferencia del fotolito. Para este fin se usa el **Persulfato de sodio** a una temperatura de 40°C. Una vez preparada la mezcla adecuada, se introduce la placa

de cobre en una cubeta con agitación y se deja actuar un tiempo de aproximadamente 10 minutos. Transcurrido este tiempo, se observa que el cobre sobrante ha desaparecido, quedando únicamente el tóner que se había transferido a la placa.

Después de realizar el ataque químico, se elimina el tóner que ha quedado depositado en la placa haciendo uso de acetona. Por último, se continúa al proceso de soldadura.

### 5.2.3. Soldadura de las placas de circuito impreso

Una vez fabricada la placa de circuito impreso, se debe realizar la soldadura de los componentes. Antes de ello, hay que limpiar correctamente la placa de cobre haciendo uso de una bañera de lavado por ultrasonidos con alcohol isopropílico.

Para la soldadura, existen dos técnicas: la soldadura con soldador manual y la soldadura con horno. Típicamente la soldadura manual se realiza para componentes THT y la soldadura con horno se emplea para componentes SMT. A continuación se explican estas dos técnicas por separado:

- **Soldadura manual:** Para la soldadura manual se deben colocar los componentes en la placa en primer lugar. Una vez colocados todos los componentes, se pone el estaño en contacto con una de las patas del componente y con el soldador (a una temperatura de 300°C). En este momento, se retira el soldador haciendo que el estaño se enfríe y el componente quede soldado.
- **Soldadura con horno:** Para la soldadura con horno se utiliza *pasta de soldadura*. Esta pasta es una mezcla de estaño y otros compuestos que hacen que sea más líquida y se facilite su aplicación. Se procede a aplicar esta pasta en los *pads* haciendo uso de una jeringuilla y su aplicador, para posteriormente posicionar los componentes sobre la PCB. Después de colocar todos los componentes en su lugar, se introduce la placa en el horno. Se selecciona el perfil de temperatura adecuado y se confirma la selección. Pasados 6 o 7 minutos se retira dicha PCB del horno, finalizando así este proceso de soldadura.

## 5.3. Fabricación del módulo CAN2Serial

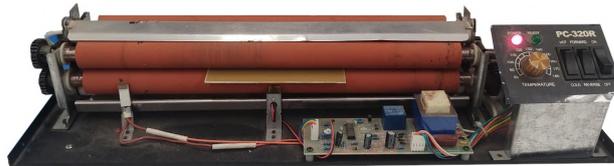
Para mostrar el proceso de fabricación se toma como ejemplo la fabricación del módulo CAN2Serial. Se ha seguido el procedimiento explicado en el apartado 5.2. En los siguientes apartados se documenta el procedimiento seguido acompañado de imágenes de cada etapa.

### 5.3.1. Proceso de fabricación de la PCB

La fabricación de la placa de circuito impreso se ha realizado utilizando el proceso combinado de atacado químico y fabricación con CNC.



(a) Taladrado con CNC.



(b) Transferencia del fotolito.

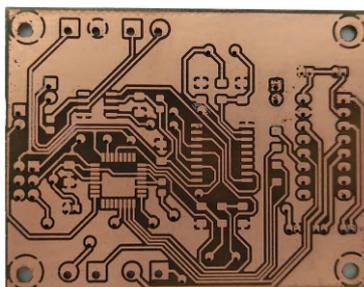
Figura 5.1: Etapas del proceso de fabricación.

Para su fabricación, se parte de una placa de cobre de doble cara sin ningún tipo de tratamiento especial como puede ser, fotosensibilidad. Por otro lado, son necesarios los ficheros de fabricación detallados en la sección 5.2.1 (tanto los GERBER con sus posteriores *G-Code* como los *.svg* de las capas de cobre). Se posiciona la placa de cobre en la base de la CNC sujetándola con cinta de doble cara y se realiza el taladrado y corte de la placa, como se muestra en la Figura 5.1a. Tras esto, se retira el borde sobrante de la placa y se limpia con ayuda de alcohol isopropílico para eliminar los residuos del corte.

Se procede a imprimir los fotolitos en papel de *transfer*, se alinean y se coloca la placa en su interior para su transferencia haciendo uso de una plastificadora de rodillos como la mostrada en la Figura 5.1b. Finalmente, se retira el fotolito.



(a) Atacado químico.



(b) PCB al eliminar el tóner.



(c) Revisión.

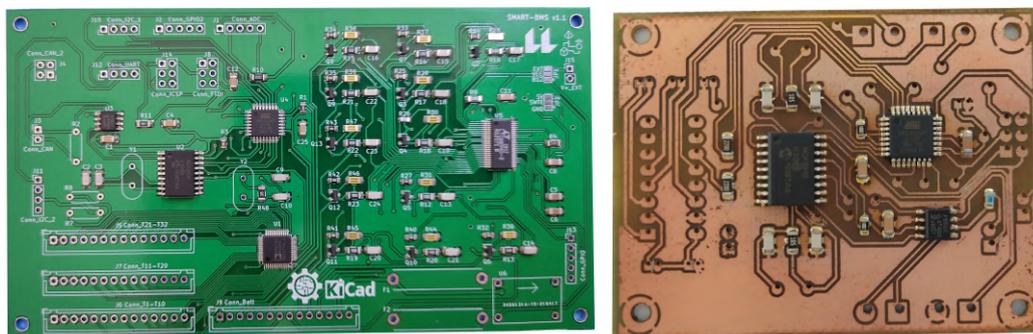
Figura 5.2: Atacado químico y revisión de la PCB.

Tras finalizar la transferencia del tóner, se prepara la mezcla de **Persulfato de sodio** con la que se realiza el atacado. Se introduce la placa en la cubeta y se procede al atacado, como se muestra en la Figura 5.2a. Una vez finalizado, se enjuaga y se aplica acetona para eliminar el tóner transferido, obteniendo finalmente la PCB con las pistas trazadas que se muestra en la Figura 5.2b.

Una vez retirado todos los restos de tóner de la PCB, se observan los posibles defectos de fabricación haciendo uso de una lupa (Figura 5.2c). En caso de existir pistas unidas o que el cobre no se haya eliminado por completo en algunas zonas, puede hacerse uso del bisturí para realizar manualmente un corte entre las pistas que se pretendan separar. Este defecto suele suceder por un tiempo de ataque químico insuficiente. En esta misma etapa, se realiza una comprobación de la correcta conductividad de todas las pistas y conexiones de la placa haciendo uso de un multímetro.

### 5.3.2. Soldadura de los componentes

Para la soldadura de los componentes en las placas, se ha hecho uso de la técnica manual y la soldadura con horno, ambas vistas en el apartado 5.2.3. En primer lugar se sueldan aquellos componentes que necesiten horno y posteriormente los que se sueldan manualmente. Para la soldadura con horno se aplica pasta de soldadura sobre los *pads* donde se colocan los componentes. En la Figura 5.3 se muestran los componentes colocados en la PCB del CAN2Serial y SmartBMS con la pasta de soldadura aplicada. Posteriormente, se introduce la placa en el horno durante 7 minutos con un perfil de temperatura gradual hasta alcanzar un máximo de 250°C. Puede verse una imagen del resultado tras haberse finalizado la soldadura con horno en la Figura 5.4.



(a) Módulo BMS.

(b) Módulo CAN2Serial.

Figura 5.3: Componentes SMD colocados.

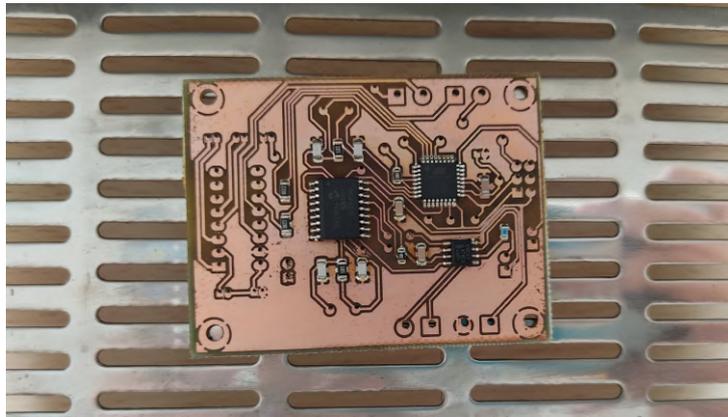
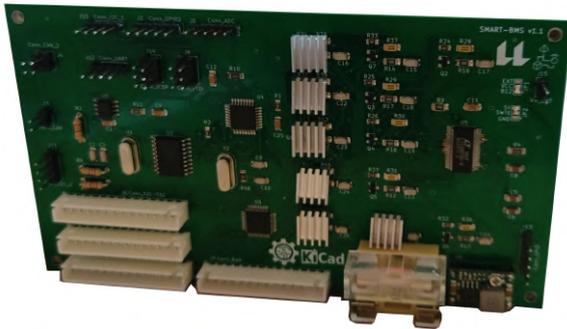
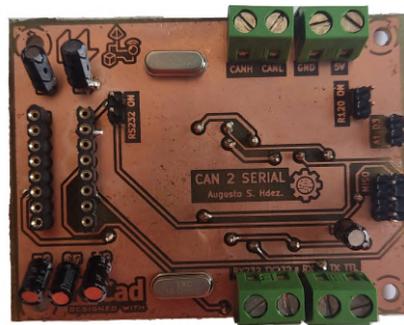


Figura 5.4: Resultado de la PCB tras soldar los componentes SMD.

Finalmente se procede a realizar la soldadura de los componentes THT (conectores, cuarzo y ciertas resistencias) haciendo uso del método de soldadura manual (Figura 5.5) y la aplicación de la laca protectora.



(a) Módulo BMS.

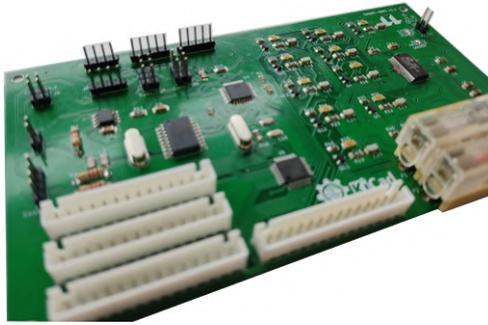


(b) Módulo CAN2Serial.

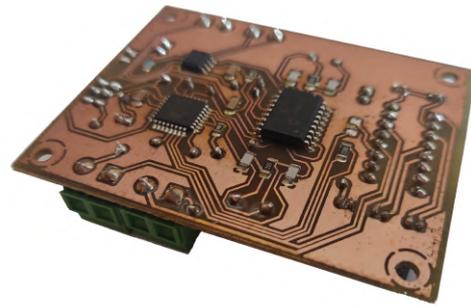
Figura 5.5: Componentes THT soldados mediante soldadura manual.

### 5.3.3. Resultado final

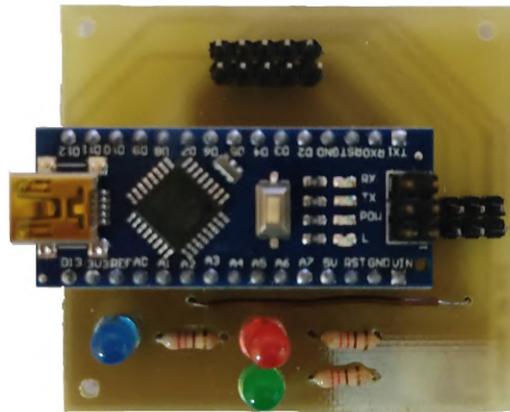
Una vez completados todos los pasos para la fabricación de los diferentes módulos, el resultado final es el mostrado en las imágenes de la Figura 5.6.



(a) Módulo BMS.



(b) Módulo CAN2Serial.



(c) Programador ICSP.

Figura 5.6: Acabado final de las PCBs.

# Capítulo 6

## Diseño y Fabricación de la batería

Como se comentó en el Capítulo 4, el BMS se empleará en una batería de diseño propio, a la que se le añade una estructura mecánica de soporte y protección, también de diseño propio. Por ello, en este capítulo se muestra el dimensionado, diseño y fabricación de la batería y su estructura de soporte, que puede ser fabricada haciendo uso de los métodos de prototipado rápido como la impresión 3D.

### 6.1. Diseño de la batería

En esta sección se muestra el cálculo y diseño de la batería además de la metodología seguida. Para ello, en primer lugar se deben caracterizar los consumos del sistema a alimentar, la tensión de suministro, así como la energía necesaria para su recarga. Posteriormente, se procede al diseño mecánico de dicha batería y su estructura protectora.

#### 6.1.1. Caracterización del sistema

El primer paso para el diseño de un sistema de abastecimiento energético es la caracterización de la carga a alimentar, estableciendo su consumo típico, las horas de uso y las fuentes de producción de energía. Para ello, se ha construido un sistema de adquisición de características similares al que se prevé alimentar en futuros proyectos y se le ha dispuesto de un *software* para la captura de sus datos de consumo energético.

El sistema a alimentar, denominado *SmartNode* (Figura 6.1), está formado por una **Raspberry Pi 3B+** con una tarjeta de adquisición **Hi-Fi Berry ADC+DAC** [32] que captura los datos de un hidrófono modelo **Aquarian AS-1** [31], alimen-

tada a 12V a través de una UPS<sup>1</sup> modelo **Juice4halt UPS** [41], al que se le ha añadido un sistema de medición de consumos formado por el sensor de corriente y voltaje **INA219** [34], con comunicación *I2C*. Se ha diseñado un servicio del sistema ejecutado en la Raspberry, que a intervalos de 10 minutos realiza la medición de la tensión y corriente de alimentación, almacenando los datos en un fichero *.json* diario y estructurando estos ficheros según el día, mes y año en que fue generado. Este sistema, se explica con mayor detalle en el Capítulo 8.

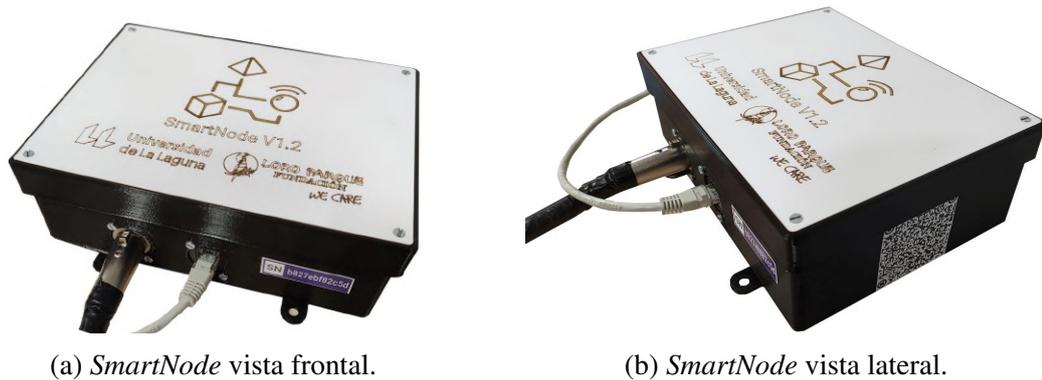


Figura 6.1: Nodo de captura de datos *SmartNode* fabricado.

Este sistema, una vez fabricado, ha sido instalado en las piscinas de *Orca Ocean* del Loro Parque. Tras varios días de uso, se han extraído los datos y se han procesado. A continuación, en la Figura 6.2 se muestran los datos del consumo de corriente de los días 22 y 23 de abril de 2021. Estos datos han servido de referencia para el cálculo de los consumos del sistema. En el Anexo C puede verse con mayor detalle todas las gráficas que a continuación se exponen.

También se han obtenido los datos de la tensión y la potencia consumida para los mismos períodos de tiempo, mostrándolos en la Figura 6.3 y 6.4. Como se puede observar, la potencia promedio consumida es de 3.2 W con una potencia máxima de 4.3 W, que alimentado a 12 V se corresponde con 267 y 358 mA, respectivamente.

Considerando un consumo continuo del sistema de 24 horas al día, la energía consumida en un día es la mostrada en la Ecuación 6.1.

$$E = V \cdot I \cdot t = 12V \cdot 0,267A \cdot 24h = 76,896 Wh \quad (6.1)$$

Teniendo en cuenta un coeficiente de seguridad de 2 en previsión de añadirle elementos al sistema de adquisición, como puede ser un sistema de telecomunicaciones basado en LoRa[14] o redes móviles, se tiene:

<sup>1</sup>*Uninterruptable Power Supply*, es un dispositivo que permite tener un flujo de energía mediante baterías o condensadores cuando el suministro eléctrico falla.

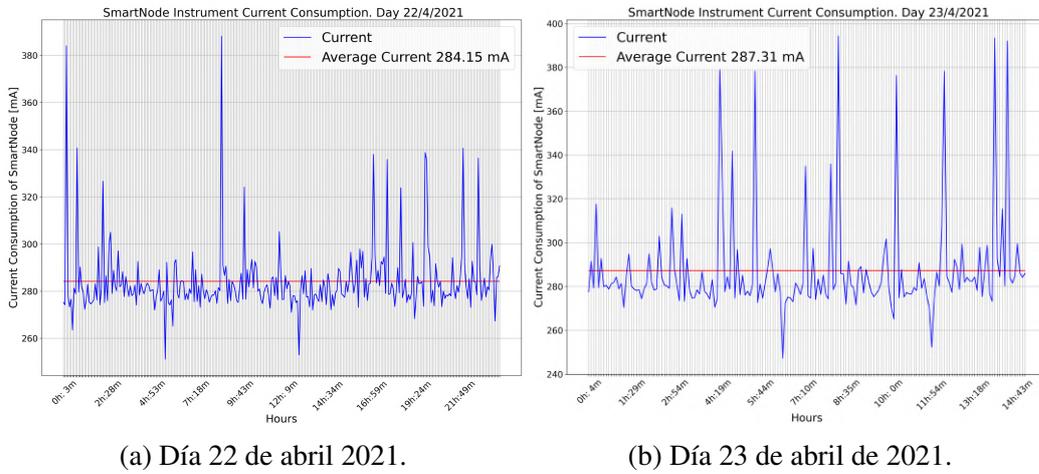


Figura 6.2: Consumos de corriente del sistema (ver Anexo C.1 y C.5).

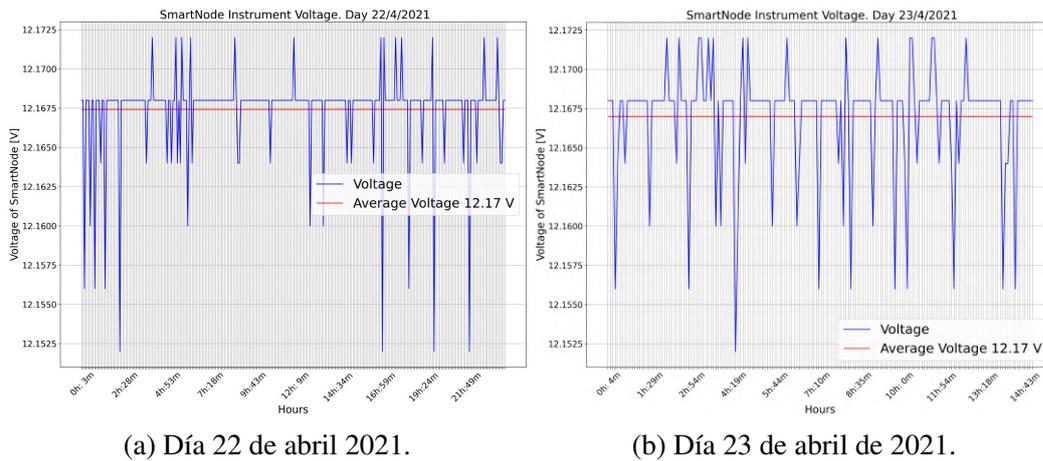


Figura 6.3: Tensión del sistema (ver Anexo C.3 y C.7).

$$E = 2 \cdot V \cdot I \cdot t = 2 \cdot 12V \cdot 0,267A \cdot 24h = 153,8 Wh \quad (6.2)$$

### Cálculo de la capacidad de la batería

Una vez obtenido el consumo de energía diario del sistema, se procede a aplicar las recomendaciones del IDAE [33]. Por tanto, se debe dimensionar la batería para abastecer 3 días completos al sistema, es decir:

$$E = 3 \cdot 153,8Wh = 461,4 Wh \quad (6.3)$$

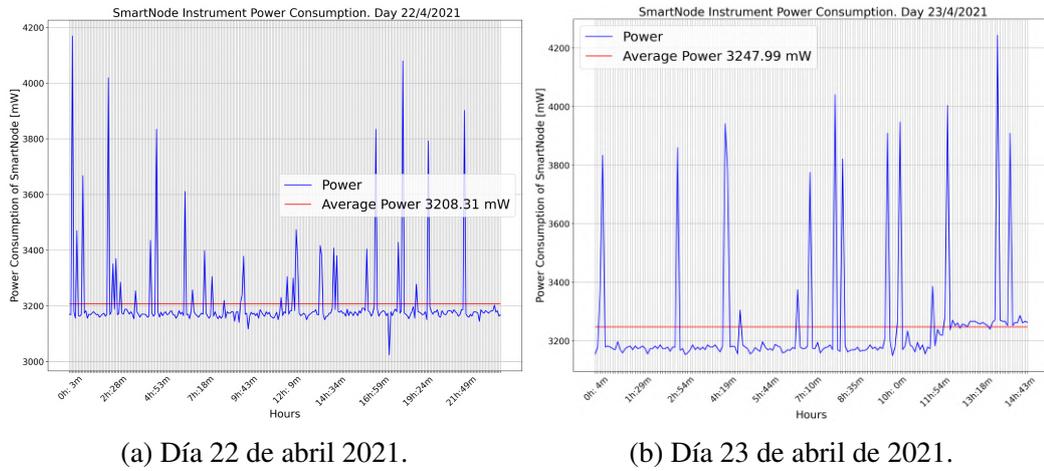


Figura 6.4: Consumo de potencia del sistema (ver Anexo C.2 y C.6).

Con el objetivo de aumentar la vida útil de la batería se le permite únicamente una profundidad de descarga diaria del 45% y estacional del 60%, obteniéndose las capacidades resultantes en las Ecuaciones 6.4 y 6.5.

$$C_{diaria} = \frac{1}{0,45} \cdot \frac{E_{consumida}}{V_{nominal}} = \frac{1}{0,45} \cdot \frac{461,4Wh}{12V} = 85,44 Ah \quad (6.4)$$

$$C_{estacional} = \frac{1}{0,6} \cdot \frac{E_{consumida}}{V_{nominal}} = \frac{1}{0,6} \cdot \frac{461,4Wh}{12V} = 64,08 Ah \quad (6.5)$$

No obstante, como el sistema generalmente operará a una tensión ligeramente superior a 12V, se decide diseñar una batería de 84 Ah.

### Cálculo del número de celdas en serie y paralelo

Como se comentó en el Capítulo 2, se hace uso de las celdas de ion-Litio de **Samsung INR21700** de 4Ah de capacidad, con una tensión nominal de 3.6V. Se procede en primer lugar, al cálculo del número de celdas a conectar en paralelo.

Como la capacidad total debe ser igual o superior a 84 Ah, se tiene que la cantidad de celdas en paralelo de cada paquete debe ser 21 (como muestra la Ecuación 6.6).

$$N_{cell_{paralelo}} = \frac{C_{total}}{C_{celda}} = \frac{84Ah}{4Ah} = 21 \quad (6.6)$$

Tras esto, se procede a calcular el número de paquetes en serie necesarios. Como el sistema opera a 12V nominales y la tensión de las celdas oscila entre los 2.6V y los 4.2V según su estado de carga, se decide emplear 4 grupos en serie,

en lugar de 3, por proporcionar un rango de tensiones siempre superior a los 12V nominales (Ecuación 6.7).

$$N_{cell_{serie}} = \frac{V_{nominal_{bateria}}}{V_{celda}} = \frac{12V}{3,6} = 3,33 \rightarrow 4 \quad (6.7)$$

Por tanto, la configuración elegida es una batería de 4 grupos en serie de 21 celdas en paralelo, lo que implica un total de 84 celdas **INR21700**. Con estas celdas, el rango de tensiones de la batería oscila entre los 10.8V (cuando todas las celdas están a su SOC del 0%) y los 16.8V con todas las celdas al 100%. Considerando la profundidad de descarga del 45% se estima una tensión mínima de la batería de 14.4V.

Esta configuración de 84Ah puede entregar una energía nominal de 1.2 kWh, con una corriente a 1C de 84A. No obstante, para las aplicaciones a las que se ha diseñado esta batería la corriente de descarga máxima será de 1 o 2 Amperios, prestándose especial interés a la alta autonomía que proporciona este sistema.

En la Figura 6.5 se muestra el esquemático de la disposición de las celdas que forman la batería.

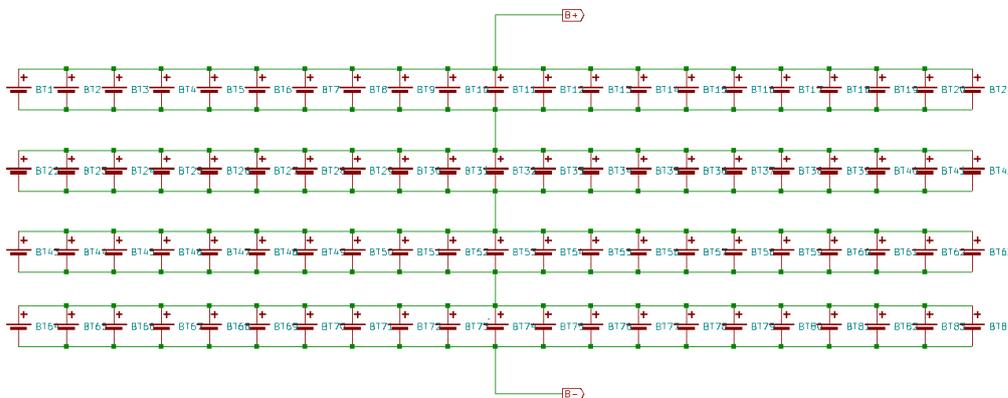


Figura 6.5: Esquema de configuración de la batería (ver Anexo F.2).

Dado que la carga de estas celdas debe realizarse a corrientes no superiores a 0.5C para evitar un desgaste prematuro de la batería, la corriente de carga máxima debe ser de 36A, aplicando un coeficiente de seguridad de 0.9. Se propone, por tanto, el uso de un cargador de 30A (como se comentó en el Capítulo 3).

Con una corriente de 30A, cada celda recibe una corriente de 1.43A (0.36C). Teniendo en cuenta que a una carga estándar de 0.5C, se toma un tiempo de 180 minutos, se prevé que a esta corriente tarde 250 minutos, aproximadamente.

### Determinación de la curva del SOC

Un aspecto importante de la batería es conocer el estado de carga de la misma. Con ello, es posible conocer con exactitud la autonomía y el tiempo de carga necesario. Existen diferentes métodos para calcular el estado de carga, destacando principalmente el **Coulomb Count**<sup>2</sup> y el método de obtención de la **Tensión de Circuito Abierto**. En este BMS diseñado, se hace uso del método de Tensión de Circuito Abierto.

El método de cálculo del SOC a partir de la tensión de circuito abierto, consiste en la medición de la tensión y la corriente de la celda, y conociendo la resistencia interna de la celda, se obtiene la tensión de circuito abierto ( $V_{oc}$ ). A partir de dicha tensión, consultando la curva de  $SOC - V_{oc}$ , es posible obtener el estado de carga.

Sin embargo, el *datasheet* de la celda no proporciona dicha curva. Por ello, se ha debido de caracterizar la celda en el laboratorio. En primer lugar, se debe construir el modelo equivalente de la celda (Figura 6.6a). Haciendo uso de la carga electrónica Rigol se ha tomado una celda totalmente cargada (SOC del 100%) y se le ha realizado una descarga muy breve a un valor de resistencia conocido. Midiendo la tensión en los extremos de la resistencia y la corriente consumida, se plantea el circuito equivalente de la celda y la carga (Figura 6.6b). Tras esto, se ha procedido a descargar la celda a una corriente constante de 800mA, midiendo el voltaje en el extremo de la carga nuevamente.

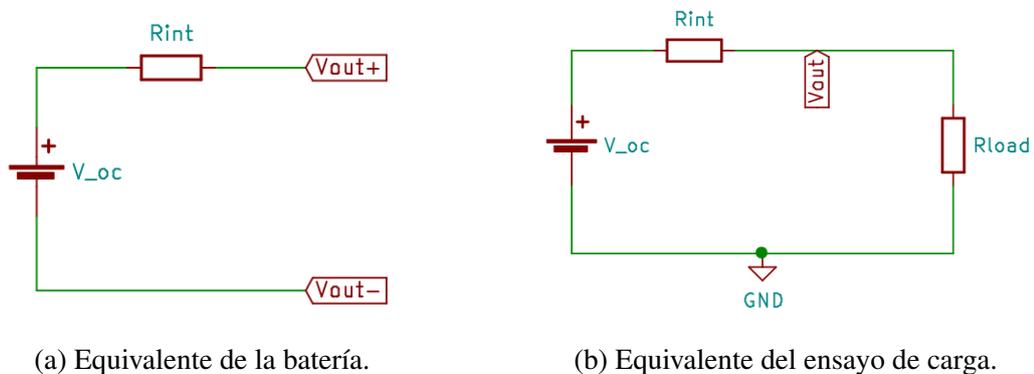


Figura 6.6: Circuitos equivalentes de la batería.

Para obtener la resistencia interna, se midió la tensión de la celda antes de conectar la carga, obteniendo una tensión de circuito abierto de 4.2V. Tras esto, se le conectó una resistencia de  $10 \Omega$  configurada en la carga electrónica, midiendo una corriente de 419.7 mA, por lo que la resistencia estimada es:

<sup>2</sup>El método de estimación del SOC mediante *Coulomb Count* consiste en, a partir de un valor de capacidad conocida, medir la corriente que circula a lo largo del tiempo para obtener la capacidad extraída de la batería y con ello calcular el SOC.

$$R_{int} = \frac{V_{oc}}{I} - R_{load} \quad (6.8)$$

$$R_{int} = \frac{4,2V}{0,4197A} - 10\Omega = 0,0071\Omega = 7,1 m\Omega$$

Este ensayo se repitió con otras dos celdas diferentes obteniéndose valores de resistencia bastante similares. Por ello, se asume como hipótesis que este valor medido es constante para todas las celdas y en todo el rango de temperaturas y tasas de descarga.

Con dicho valor calculado, es posible escribir las ecuaciones del modelo equivalente de la batería. Considerando positivo el sentido de la corriente entrando hacia la batería (cargando) y negativo saliendo (descargando), se tiene:

$$I_{out} = \frac{V_{oc}}{R_{int} + R_{load}} \quad (6.9)$$

$$V_{oc} = V_{out} - I \cdot R_{int} \quad (6.10)$$

$$V_{oc} = V_{out} - I \cdot 0,0071$$

Por otro lado, es posible obtener la capacidad extraída de la batería (en Ah) a partir de la integración de la corriente consumida en el tiempo, aplicando el método de *Couloumb Count*, como se muestra a continuación en la Ecuación 6.11.

$$C_{out} = \int_{t_o}^{t_f} i(t) dt \quad (6.11)$$

Sin embargo, como se captura un dato de corriente cada 10 segundos, no es posible aplicar directamente la integral, sino que esta debe discretizarse, quedando como se muestra en la Ecuación 6.12, siendo  $\Delta t$  incrementos de 10 segundos.

$$C_{out} = \sum_{j=0}^{j=N} i[j \Delta t] \Delta t \quad (6.12)$$

Una vez conocida la capacidad extraída de la batería en cada instante, puede calcularse la capacidad restante en ese instante como la diferencia entre la capacidad inicial (obtenida como la capacidad en el último instante del ensayo) y la capacidad extraída. Con estos datos, puede calcularse también el estado de carga como se muestra en la Ecuación 6.13.

$$C_{bateria} = C_{inicial} - C_t \quad (6.13)$$

$$SOC = \frac{C_{inicial} - C_t}{C_{inicial}} \quad (6.14)$$

Este ensayo se realizó nuevamente con tres celdas diferentes a una corriente de descarga de 800mA, obteniéndose un conjunto de datos que fueron procesados aplicando la metodología explicada anteriormente. La curva del estado de carga frente a la tensión de circuito abierto se muestra en la Figura 6.7.

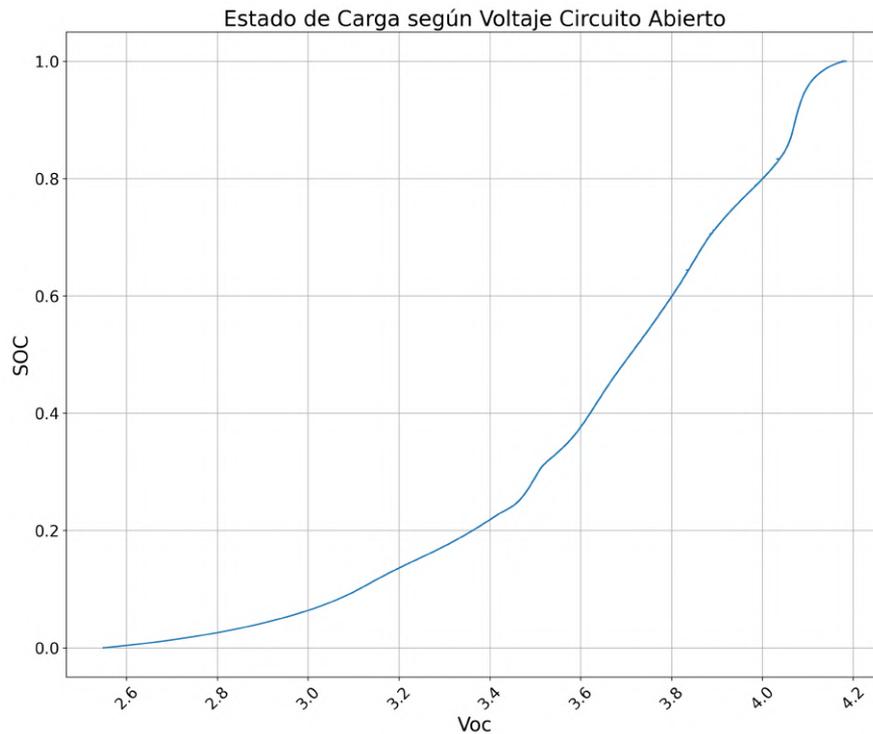


Figura 6.7: Curva del SOC (Anexo B.1).

Tras esto, se procedió a realizar una regresión polinómica buscando la curva que mejor definiera el comportamiento de los valores del ensayo para ser programada en el *firmware* del BMS. Se iteró con curvas de hasta grado 8, siendo la que mejor se ajusta la de grado 4, con un error relativo máximo del 3.7%. La curva seleccionada se muestra en la Figura 6.8. No obstante el resto de curvas pueden ser visualizadas en el Anexo B de este proyecto.

La función de grado 4 que describe la curva de la figura es la siguiente:

$$SOC = -0,4203 \cdot V_{oc}^4 + 5,6962 \cdot V_{oc}^3 - 28,2351 \cdot V_{oc}^2 + 61,1830 \cdot V_{oc} - 49,1338 \quad (6.15)$$

Sustituyendo la tensión de circuito abierto por su expresión, se tiene:

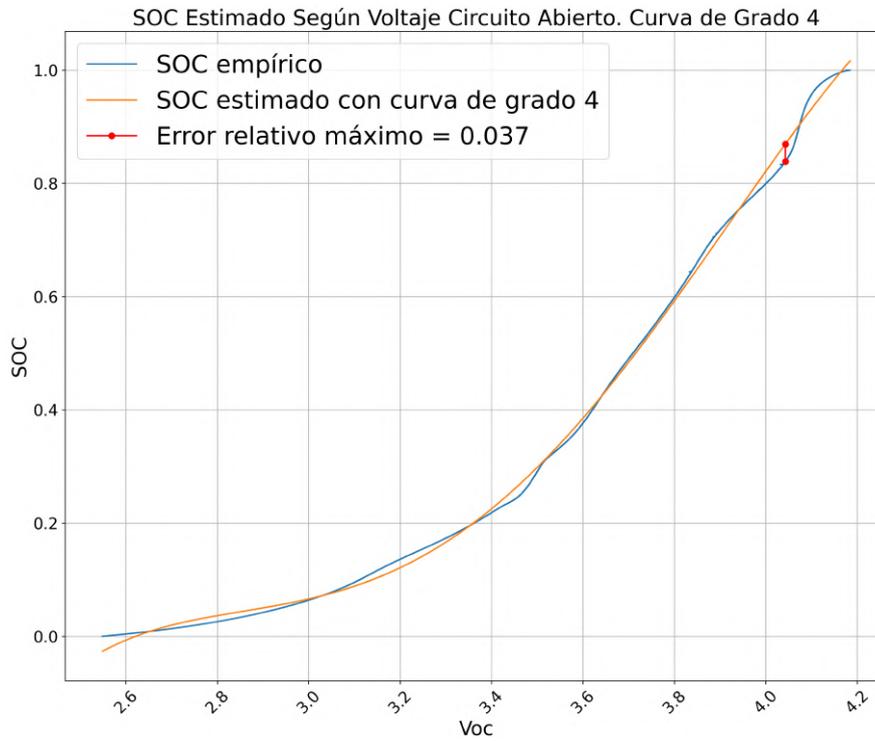


Figura 6.8: Curva del SOC empírica y calculada con grado 4 (Anexo B.5).

$$\begin{aligned}
 SOC = & -0,4203 \cdot (V_m - I_m \cdot R_{int})^4 + 5,6962 \cdot (V_m - I_m \cdot R_{int})^3 \\
 & - 28,2351 \cdot (V_m - I_m \cdot R_{int})^2 + 61,1830 \cdot (V_m - I_m \cdot R_{int}) - 49,1338
 \end{aligned} \quad (6.16)$$

Este ensayo se repitió con otras dos celdas, obteniendo las siguientes funciones:

$$\begin{aligned}
 SOC = & -0,4220 \cdot V_{oc}^4 + 5,7138 \cdot V_{oc}^3 - 28,2942 \cdot V_{oc}^2 + 61,2542 \cdot V_{oc} - 49,1400 \\
 SOC = & -0,4219 \cdot V_{oc}^4 + 5,7138 \cdot V_{oc}^3 - 28,2942 \cdot V_{oc}^2 + 61,2542 \cdot V_{oc} - 49,1399
 \end{aligned} \quad (6.17)$$

Se han promediado los coeficientes de los tres ensayos, obteniendo la expresión que se muestra en la Ecuación 6.18 como expresión de la curva  $SOC - V_{oc}$ .

$$SOC = -0,4214 \cdot V_{oc}^4 + 5,7079 \cdot V_{oc}^3 - 28,2745 \cdot V_{oc}^2 + 61,2305 \cdot V_{oc} - 49,1379 \quad (6.18)$$

### 6.1.2. Diseño mecánico de la batería

En el presente apartado se muestra el diseño de las piezas a fabricar para realizar las conexiones de la batería, así como la estructura de soporte y protección.

#### Diseño de un *pack* de celdas

Con el objetivo de diseñar una batería que sea modular, pudiendo aumentarse tanto en capacidad como en voltaje, se ha diseñado un sistema formado por pequeños paquetes intercambiables y apilables horizontalmente. De esta forma se simplifican las tareas de mantenimiento y se le dota de flexibilidad, pudiendo sustituir únicamente el paquete en el que se encuentre la celda o celdas dañadas en lugar de toda la batería. Además, con el diseño propuesto puede construirse una batería de cualquier tensión nominal (en múltiplos de 3.7V) con grupos de celdas múltiplo de 21.

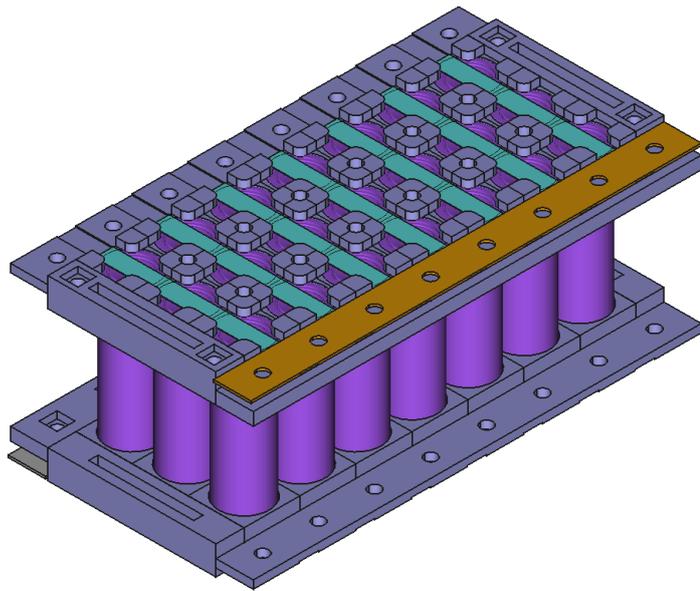


Figura 6.9: Estructura básica de un *paquete*.

La batería está formada por 4 paquetes de 21 celdas unidos entre sí mediante tornillos. Cada paquete está formado por 3 hileras de 7 celdas en paralelo. Estas están unidas todas entre sí mediante pletinas de **níquel de 8 mm de ancho y 0.2 mm de espesor**, que se conectan a un **busbar de cobre de 16 mm de ancho y 0.8 mm de espesor**, que está en contacto con el busbar del siguiente paquete. Los tornillos además de realizar la unión de los paquetes, garantizan el buen contacto

entre las pletinas de cobre de los paquetes conectados en serie. La Figura 6.9 muestra la estructura básica que forma un paquete de 21 celdas y 3.7V.

Dado que los *soportes de celdas* se han diseñado para ser impresos en 3D con filamento PET-G<sup>3</sup> o similar, se han añadido huecos para embutir las tuercas cuadradas M5 en los *salientes* de la estructura, de modo que los tornillos aprieten contra la estructura impresa, garantizando así el buen contacto de ambas láminas de cobre. En dichos salientes, se han dispuesto unas ranuras de  $8\text{ mm} \times 0.2\text{ mm}$  para permitir el alojamiento de las pletinas de níquel. Además, se han añadido huecos para las mismas tuercas entre las celdas, permitiendo así sujetar estos paquetes a una caja o carcasa o emplear varilla roscada para rigidizar el *pack*.

Los paquetes inicial y final, están sujetos por unas *tapetas laterales* que dotan de rigidez al sistema y permiten la conexión del BMS en uno de sus extremos. Se muestran en la Figura 6.10. Estas piezas, se unen a la estructura de los paquetes empleando el mismo sistema de unión *tornillo-tuerca cuadrada*.

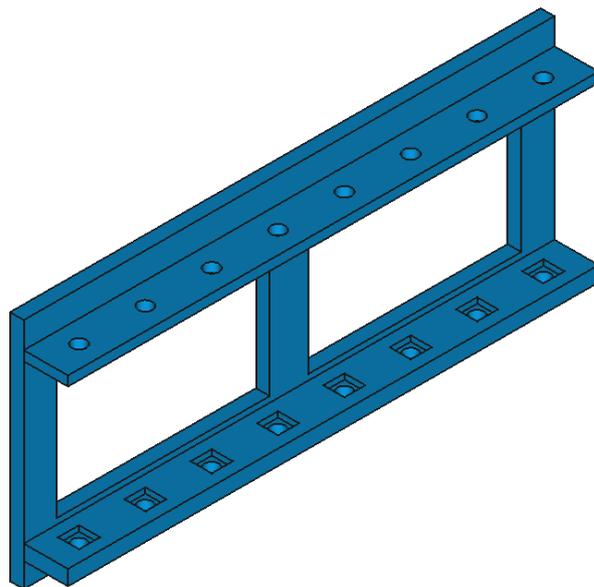


Figura 6.10: Tapas laterales de los paquetes

---

<sup>3</sup>El copoliéster de polietileno tereftalato glicol es un material plástico rígido, con una alta resistencia térmica y mecánica ampliamente utilizado en la industria alimentaria. Presenta unas características mecánicas mejoradas con respecto al PET.

### Dimensionado de los contactos de la batería

Como se comentó anteriormente, las celdas estarán unidas entre sí mediante pletinas de níquel de 8 mm de ancho, 0.2 mm de espesor y 88 mm de largo en grupos de 3 y posteriormente todas entre sí mediante una pletina de cobre de 0.8 mm de espesor y 16 mm de ancho. Para determinar estas dimensiones se ha tenido en cuenta una corriente máxima admisible de 84A, con una temperatura máxima de 40°C y una caída de tensión entre los bornes de la batería inferior al 1.5 %.

En primer lugar, se dimensionan las láminas de Níquel. Teniendo en cuenta que el paquete de 21 celdas, a 1C, puede proporcionar 84A, por cada pletina de níquel circula 1/7 de la corriente, es decir, 12A. Con las dimensiones de la pletina, se calcula su resistencia a 25°C y a 45°C, obteniendo:

$$R_{Niquel} = [\rho_o \cdot (1 + \alpha \cdot \Delta t)] \cdot \frac{L}{s}$$

$$R_{@25C} = [6,4 \cdot 10^{-8}(\Omega m) \cdot (1)] \cdot \frac{88 \cdot 10^{-3}}{8 \cdot 10^{-3} \cdot 0,2 \cdot 10^{-3}} = 0,00352 \Omega$$

$$R_{@45C} = [6,4 \cdot 10^{-8}(\Omega m) \cdot (1 + 0,005(\Omega/^{\circ}C) \cdot 20^{\circ}C)] \cdot \frac{88 \cdot 10^{-3}}{8 \cdot 10^{-3} \cdot 0,2 \cdot 10^{-3}} =$$

$$= 0,003872 \Omega$$
(6.19)

Con el valor de resistencia y corriente máxima se calcula la caída de tensión en un paquete debida a las pletinas de níquel, teniendo en cuenta que la resistencia equivalente será la asociación en paralelo de las 7 pletinas:

$$\Delta V = I_{max} \cdot \frac{R_{@45^{\circ}C}}{7} = 84A \cdot \frac{0,003872\Omega}{7} = 0,046 V \quad (6.20)$$

$$\Delta V(\%) = \frac{0,046V}{3,7V} = 1,24\% \quad (6.21)$$

Tras esto, se procede a calcular la resistencia del *bus* de cobre. Para ello, se tiene en cuenta las dimensiones de la pletina y se le descuentan los 8 agujeros de 5.2 mm realizados para los tornillos (Ecuación 6.22).

$$R_{Cu} = [\rho_o \cdot (1 + \alpha \cdot \Delta t)] \cdot \frac{L}{s}$$

$$R_{@25C} = [1,71 \cdot 10^{-8}(\Omega m)] \cdot \frac{16 \cdot 10^{-3}}{(192 - 8 \cdot 5,2) \cdot 10^{-3} \cdot 0,8 \cdot 10^{-3}} = 2,2739 \cdot 10^{-5} \Omega$$

$$R_{@45C} = [1,71 \cdot 10^{-8}(\Omega m) \cdot (1 + 0,00393(\Omega/^{\circ}C) \cdot 20^{\circ}C)] \cdot \frac{16 \cdot 10^{-3}}{(192 - 8 \cdot 5,2) \cdot 10^{-3} \cdot 0,8 \cdot 10^{-3}} = 2,4527 \cdot 10^{-5} \Omega$$
(6.22)

Con ella, se obtiene la caída de tensión en el *bus* de cobre:

$$\Delta V = 84A \cdot 2,4527 \cdot 10^{-5} \Omega = 0,00206 V \quad (6.23)$$

$$\Delta V(\%) = \frac{0,00206V}{3,7V} = 0,06\% \quad (6.24)$$

Por tanto, la caída de tensión total en el paquete es del 1.3 %, inferior al 1.5 % establecido en el criterio de cálculo. Esta caída de tensión se mantendrá constante para una batería de  $n$  paquetes en serie, con 21 celdas en paralelo, dado que la tensión nominal sube en el mismo factor que la resistencia total de la batería.

### Dimensionado de los cables de la batería

Por último, se dimensiona la sección del cableado que une las barras de cobre de los extremos con los bornes exteriores de la caja de batería (Ecuación 6.25). Se considera una longitud de 15 cm para el cable positivo y negativo y se procede a determinar la sección mínima para la corriente de 84A y una caída de tensión global (sumando la anterior) inferior al 1.5 %.

$$S_{minima} = \frac{2 \cdot \rho \cdot L \cdot I}{\Delta V} = \frac{2 \cdot (1,71 \cdot 10^{-8} \cdot (1 + 0,00393 \cdot 20^{\circ}C)) \cdot 0,15m \cdot 84A}{0,02 \cdot 14,4V} =$$

$$= 3,2277 \cdot 10^{-6} m^2 = 3,23 mm^2 \rightarrow 4 mm^2$$
(6.25)

Se seleccionan cables de  $4mm^2$  de sección, para garantizar una caída de tensión en los cables inferior al 0.2 %, cantidad máxima para que la caída del conjunto no supere el 1.5 %.

### 6.1.3. Aspecto de la batería de 4 paquetes en serie

A continuación, en la Figura 6.11 se muestra el aspecto de la batería finalmente diseñada. Está formada por 4 paquetes de 21 celdas como el explicado anteriormente, conectados mediante las pletinas de cobre. Las celdas se disponen con orientación opuestas entre los paquetes, comenzando con el polo positivo orientado hacia arriba en el primer paquete (situado a la izquierda) y finalizando con dicho polo orientado hacia abajo en el cuarto paquete.

Las dimensiones de la batería son de **192 mm de ancho, 85 mm de alto y 383 mm de largo**. La resistencia interna de la batería debida a las pletinas de níquel, de cobre y los cables a 25°C y 40°C es de 0.0154  $\Omega$  y 0.0169  $\Omega$ , respectivamente.

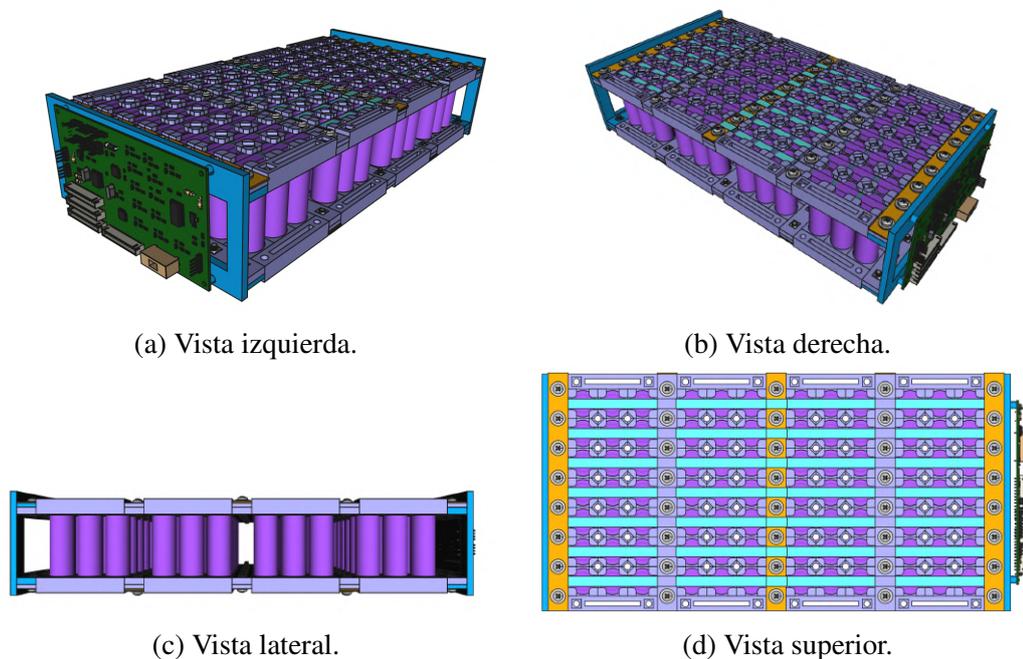
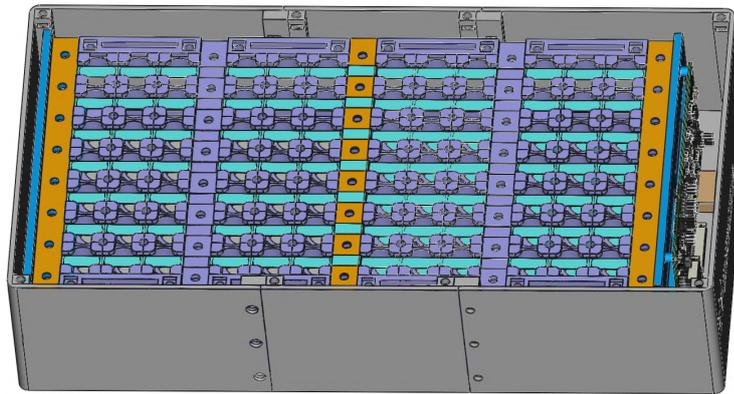


Figura 6.11: Aspecto del paquete de celdas diseñado.

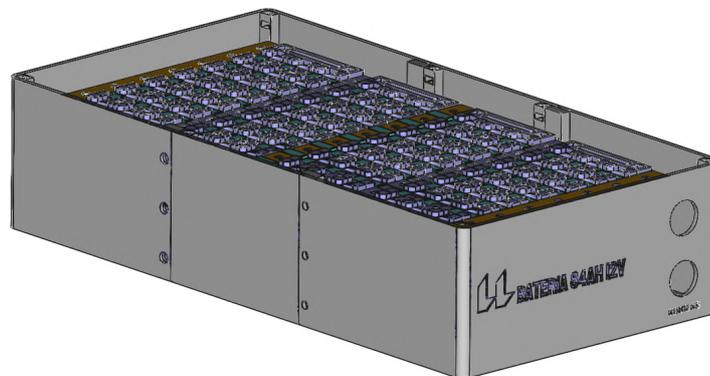
### 6.1.4. Diseño de la caja de la batería

Una vez finalizado el diseño de la estructura de la batería. Se procede a diseñar una caja para albergar la misma. Esta caja, también ha sido impresa en 3D, por ello se ha diseñado como un conjunto de piezas a ensamblar empleando el mismo método de unión *tornillo-tuerca cuadrada* que con los *packs* de baterías. Se ha diseñado una caja con tapa superior atornillable, dividida en 3 módulos imprimibles (*estructura izquierda, estructura central y estructura derecha*) que puedan

ser impresos de forma individual en la impresora *Ender 3* (por sus dimensiones el diseño no pudo hacerse como una sola pieza). Estos tres módulos se muestran ensamblados en la Figura 6.12. Como se puede observar, en el módulo derecha, además del logo *ULL* y la inscripción "*Batería 84Ah 12V*" se han practicado dos huecos de 23 mm de diámetro para instalar dos conectores IP68 de 3 pines para la comunicación CAN.



(a) Vista superior.



(b) Vista derecha.

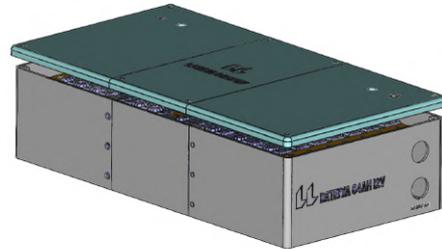
Figura 6.12: Aspecto de la base de la caja de la batería.

La tapa también ha sido dividida en 3 partes. En las tapas laterales se alojan los bornes de la batería, por lo que se han diseñado 2 orificios de 10 mm de diámetro. En la tapa central se ha realizado nuevamente la inscripción del logo de

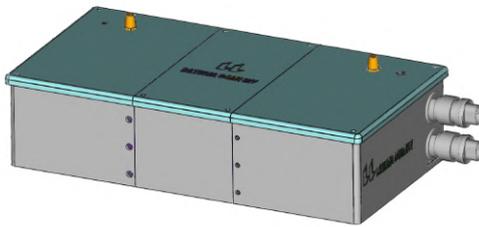
la Universidad de La Laguna, así como la capacidad de la batería. La estructura completa (caja con la tapa) puede verse en la Figura 6.13.



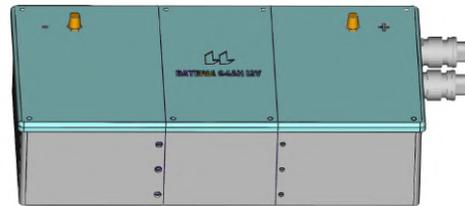
(a) Vista frontal sin conectores.



(b) Vista derecha sin conectores.



(c) Vista lateral con bornes y conectores.



(d) Vista frontal con bornes y conectores.

Figura 6.13: Aspecto de la caja de la batería.

## 6.2. Fabricación de la batería

En esta sección, se introducirán los diferentes elementos mecánicos que se han fabricado además de la metodología seguida para ello.

### 6.2.1. Características mecánicas de las piezas

En la sección 6.1.2 se describen las características técnicas que presentan los diseños de las piezas en su fase de diseño 3D. Sin embargo, también es preciso definir las características físicas que estas han de tener para poder ser fabricadas. Algunas de estas características son las siguientes:

- **Material de fabricación:** Para la fabricación de las piezas se usarán materiales plásticos con resistencia a la temperatura como el ABS<sup>4</sup> o PET-G.

<sup>4</sup>El acrilonitrilo butadieno estireno, o ABS por sus siglas en inglés (*Acrylonitrile Butadiene Styrene*), es un plástico muy resistente al impacto muy utilizado en automoción y otros usos tanto industriales como domésticos.

- **Conexión entre celdas:** Para la unión de las celdas de un mismo paquete entre sí se usarán pletinas de níquel, con una pureza superior al 98 % de 8 mm de ancho y 0,2 mm de espesor.
- **Conexión entre paquetes:** Para la conexión eléctrica de un paquete de baterías con el siguiente se usarán pletinas de cobre de 0.8 mm de espesor.
- **Sistemas de fijación:** Para la unión de los diferentes elementos se usarán tornillos DIN 7985 INOX A4 y tuercas cuadradas DIN 557 INOX A4.

Por otro lado, las piezas han sido diseñadas para ser impresas con la impresora 3D **Ender 3**, descrita en el Capítulo 2. Esta presenta unas dimensiones máximas de impresión de *20cm x 20cm x 20cm*, por lo que la caja y carcasa exterior de la batería debe ser dividida en tres piezas para poder ser impresa en esta máquina.

### 6.2.2. Metodología de fabricación de piezas

Para la fabricación de las piezas principalmente se ha usado la **impresión 3D**. No obstante, los elementos metálicos han sido cortados haciendo uso de una **cizalla** y se han realizado sus orificios con un **taladro de banco** y la broca adecuada. A continuación se explica en detalle el método de impresión 3D.

#### Impresión 3D de piezas

Para la impresión 3D de las piezas diseñadas, es necesario que los diseños estén en uno de los formatos compatibles con el *software Prusa Slic3r*; generalmente *.obj* o *.stl*. Una vez se han exportado correctamente las diferentes piezas a estos formatos, se introducen en el *slicer* para generar los ficheros *G-Code* que serán llevados a la impresora para realizar los trabajos de impresión.

En el *software* laminador se configuran los parámetros básicos de impresión, como puede ser la temperatura del *hot-end* y de la cama, el flujo de filamento, la velocidad de los movimientos, etc. en función del material con el que se va a imprimir.

Una vez configurados dichos parámetros y generados los *G-Code*, se introducen dichos ficheros en la impresora 3D haciendo uso de una tarjeta de memoria *micro SD*. Se aplica una ligera capa de laca a la cama (para mejorar su adherencia) y se procede a comenzar con la impresión de las piezas.

Finalmente, una vez finalizada la impresión, se procede a retirar la pieza haciendo uso de una espátula y se elimina la laca de la cama enjuagando la misma.

Solo resta el uso de una lija o un *cutter* y un alicate para eliminar los soportes<sup>5</sup> y para corregir pequeñas imperfecciones surgidas durante la impresión.

### 6.2.3. Fabricación de los soportes de las celdas

Para la fabricación de un paquete de celdas es necesario realizar la impresión de al menos 2 *soportes de celdas* y 2 *tapetas laterales*. Estas piezas han sido impresas en PET-G de color negro y gris para diferenciar los polos positivos y negativos del paquete (negro corresponde al negativo y gris al positivo). El filamento ha sido extruido a una temperatura de 230°C y la cama se ha establecido a 60°C. Una vez impresas, se han retirado los soportes de impresión y se ha procedido a colocar las tuercas cuadradas y las celdas INR21700, como se muestra en la Figura 6.14.

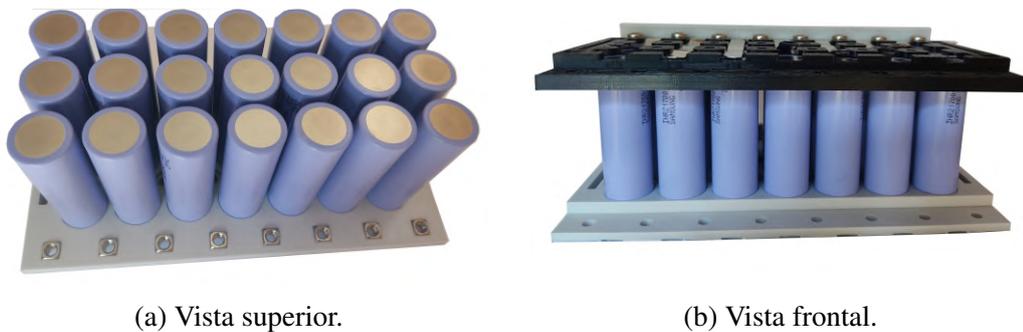


Figura 6.14: Módulo de 1 paquete de batería impreso.

### 6.2.4. Fabricación de las láminas de unión de las celdas

Para realizar la unión entre celdas se ha hecho uso de pletinas de níquel y un bus de cobre. Las láminas de níquel se han cortado a una longitud de 88 mm y las láminas de cobre a 193 mm. A estas últimas se le han realizado 8 orificios de 5.2mm de diámetro para la sujeción de las pletinas de cobre con tornillo M5 y otros 7 agujeros de 4.2mm para poder realizar la unión de las láminas de níquel con el cobre haciendo uso de estaño. En la Figura 6.15a y 6.15b se puede observar el aspecto de las uniones níquel-cobre tras la soldadura con estaño. La Figura 6.15c muestra el aspecto de dicha unión visto desde la otra cara y en la Figura 6.15d se puede ver el conjunto total de piezas fabricadas.

<sup>5</sup>Los soportes de impresión se emplean para permitir la impresión de piezas que presentan partes en voladizo, es decir, que no están en contacto con la base.

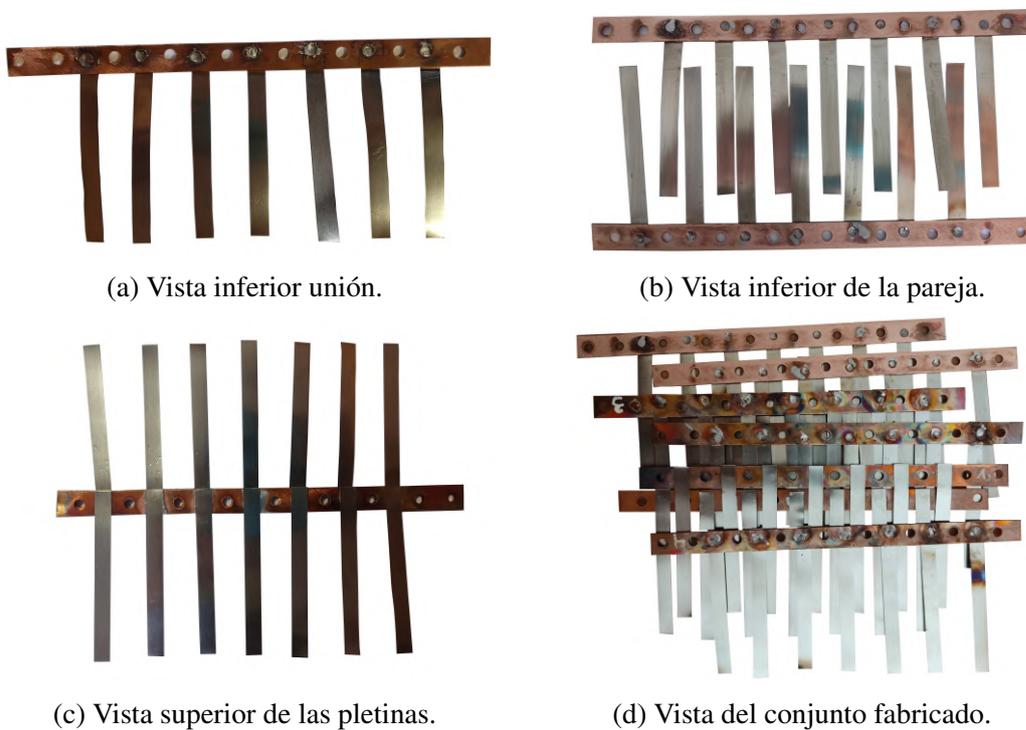


Figura 6.15: Pletinas de cobre-níquel fabricadas.

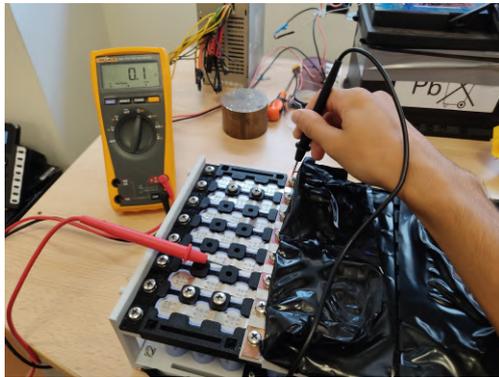
### 6.2.5. Montaje y soldadura de las pletinas

Una vez taladradas y soldadas las pletinas de níquel y cobre, se procede a realizar la unión de las celdas. Para ello se emplea la soldadura de punto. A cada celda se le han realizado 3 soldaduras.

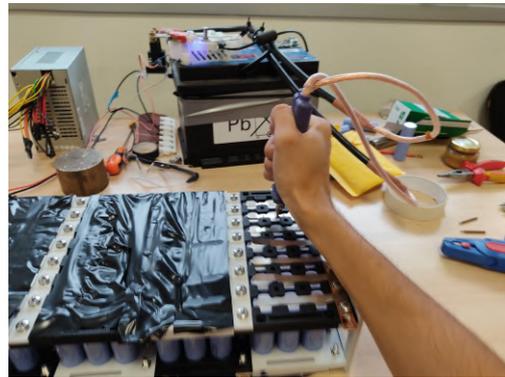
En primer lugar, se ha dispuesto el conjunto de pletinas, luego se han ido soldando celda a celda y posteriormente, se ha comprobado la tensión entre un extremo de la celda y el extremo opuesto en la barra de cobre. En la Figura 6.16 se muestran fotos del proceso de fabricación de un paquete.

### 6.2.6. Preparación del cableado

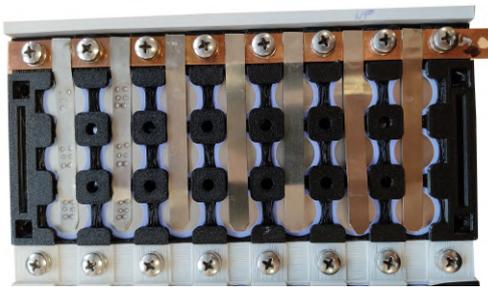
Tras montar todas las pletinas de conexión, se ha procedido a conectar los termistores y los cables de medición de las celdas. Se ha dispuesto un termistor en la pletina de cobre del positivo, otro en el negativo y otros dos distribuidos, uno en cada cara de la batería. Para la medición de tensiones se han empleado cables de  $0.75\text{mm}^2$  que se han colocado en los extremos de las pletinas de cobre empleando terminales de arandela de 4.2mm y **tornillo DIN 7985 INOX M4** con su respectiva tuerca. Tanto los cables de medición y balanceo como los termistores



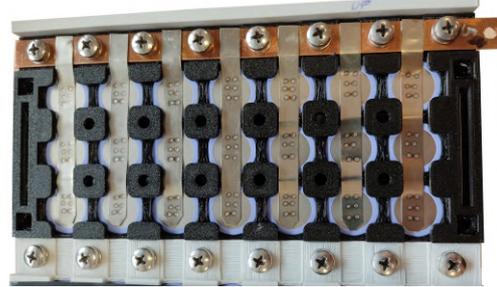
(a) Medición de celdas montadas.



(b) Proceso de soldado por punto.



(c) Paquete a medio soldar.



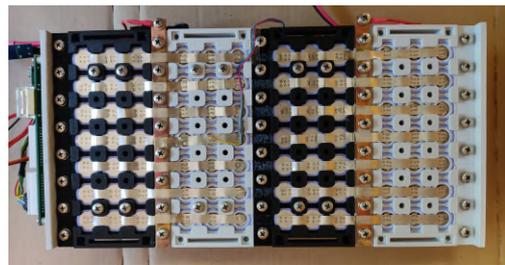
(d) Paquete soldado.

Figura 6.16: Proceso de soldadura de 1 paquete de batería.

se han llevado a conectores **JST** que se instalarán en el BMS. Este montaje puede observarse en la Figura 6.17.



(a) Vista superior del paquete.



(b) Vista inferior del paquete.

Figura 6.17: Batería completa montada con su cableado.

Posteriormente, se ha añadido la PCB del BMS (Figura 6.18a), se ha conectado todo el cableado a la placa y se han añadido los cables de salida de los bornes con un fusible de 35A (Figura 6.18b) y el sensor de corriente **ACS712**. Para pro-

tegerla de contactos accidentales, se ha colocado una lámina aislante específica para baterías.

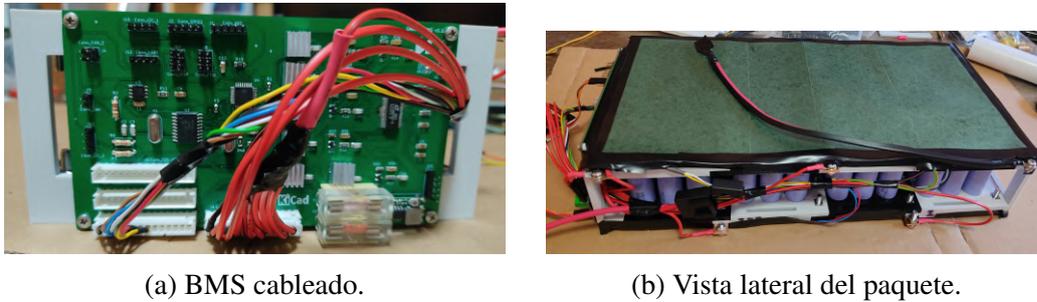


Figura 6.18: Batería montada con el BMS.

### 6.2.7. Fabricación de la caja protectora de la batería

Cuando se ha finalizado el montaje del paquete de celdas, se procede a imprimir y montar las piezas de la caja de la batería (Figura 6.19).

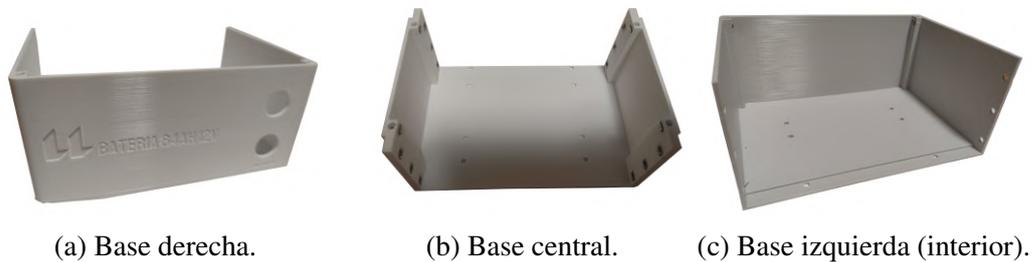
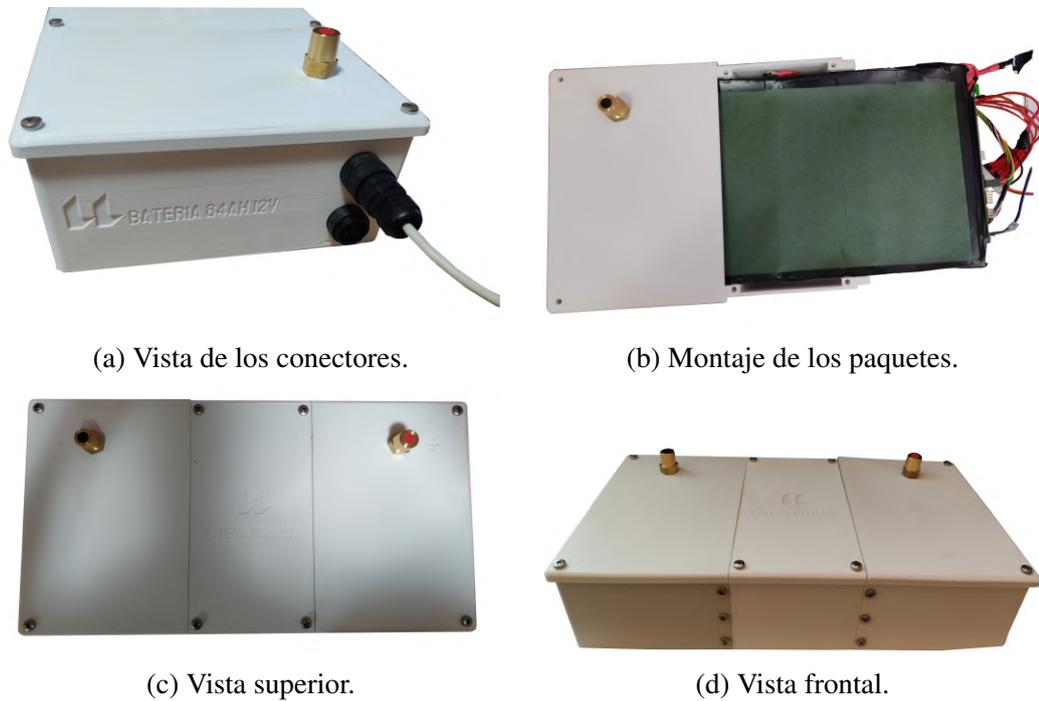


Figura 6.19: Piezas impresas de la base de la caja de batería.

En primer lugar, se insertan todas las tuercas cuadradas en sus alojamientos, se montan las tres piezas, se sitúa el paquete de celdas (Figura 6.20b), los conectores IP68 de comunicación (Figura 6.20a) y la tapa con los bornes. El aspecto final de la caja de batería cerrada, se muestra en las Figuras 6.20c y 6.20d.



(a) Vista de los conectores.

(b) Montaje de los paquetes.

(c) Vista superior.

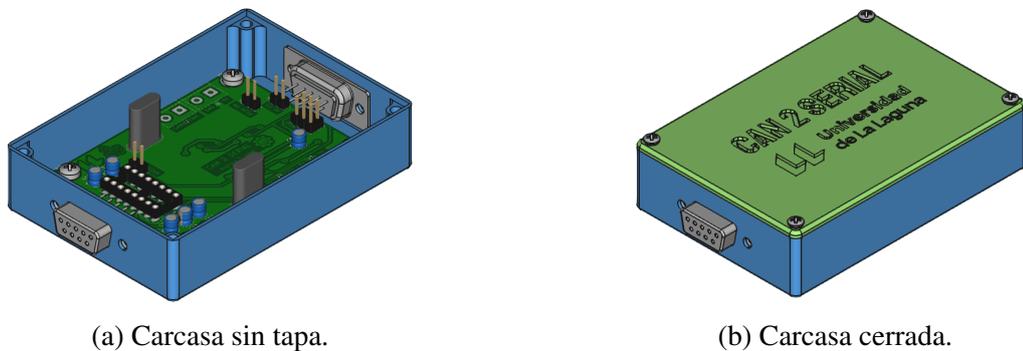
(d) Vista frontal.

Figura 6.20: Montaje de la caja de la batería.

### 6.3. Montaje y fabricación de la carcasa del CAN2Serial

Empleando la misma metodología de diseño y fabricación explicada anteriormente, se procedió a diseñar una carcasa para proteger la PCB del convertor de CAN a Serial. Esta carcasa se diseñó para alojar dos conectores DB9, uno para la comunicación CAN y otra para la conexión serial.

El diseño 3D puede observarse en las imágenes de la Figura 6.21.



(a) Carcasa sin tapa.

(b) Carcasa cerrada.

Figura 6.21: Carcasa de CAN2Serial diseñada.

Esta carcasa fue impresa en 3D empleando filamento PLA<sup>6</sup> y se montó en ella la PCB del *conversor*. Para las sujeciones se han empleados tornillos M3x10mm. La carcasa fabricada, con la PCB y los conectores DB9 montados, se observan en la Figura 6.22.



(a) Carcasa impresa.



(b) Carcasa con los cables conectados.

Figura 6.22: Carcasa de *CAN2Serial* fabricada.

## 6.4. Montaje y fabricación del *configurador portátil*

Empleando la misma metodología de fabricación explicada para el desarrollo de la carcasa del *CAN2Serial*, se procede a imprimir el diseño de carcasa para Raspberry Pi 3B+ y su pantalla táctil capacitiva de 7"[51] (Figura 6.23). En este caso, la carcasa no es un diseño propio sino que ha sido diseñada por Dan Kehoe [42] y puede descargarse desde el repositorio de diseños *Thingiverse* [23].



(a) Vista lateral.



(b) Vista trasera.

Figura 6.23: Carcasa del *configurador* fabricada.

<sup>6</sup>El ácido poliláctico (PLA) es un polímero constituido por moléculas de ácido láctico, con propiedades similares a las del tereftalato de polietileno (PET) que se utiliza en la fabricación de envases, pero que además es biodegradable.

Tras imprimirla, se ha montado en ella la Raspberry Pi 3B+ con la pantalla táctil y se ha instalado el paquete de *software* de diseño propio *BMS Programmer GUI* que se explicará posteriormente en el Capítulo 8. Con ello, es posible tener un equipo portátil de diagnóstico que, a través de la interfaz CAN2Serial, se conecte al bus CAN y permita la monitorización y configuración del BMS y demás dispositivos.

# Capítulo 7

## Cálculo del sistema de generación

En el presente capítulo se expone el dimensionado del sistema de generación necesario para abastecer de energía eléctrica a la batería y el dispositivo de consumo caracterizados en el Capítulo 6. Se ha propuesto como medio de generación la energía solar. El criterio de cálculo ha sido tomado de lo expuesto en el libro *"Instalaciones generadoras fotovoltaicas"* de Vicente Mascarós [47] y en *"Instalaciones de energía solar fotovoltaica. Pliego de condiciones técnicas aisladas de red"* [33].

### 7.1. Premisas de cálculo

Como se ha comentado anteriormente, se ha tomado como criterio para el cálculo lo expuesto en el [33] y [47]. En cuanto al procedimiento de cálculo, en primer lugar se calcula la energía consumida en el día, tras esto, se estudia la ubicación de los paneles, y se aplica el criterio del mes crítico, obteniendo una tabla de irradiación para los diferentes meses del año e inclinaciones y se procede a calcular la relación *consumo/irradiación*. Con ella, se decide la inclinación óptima y tras ello, se procede a dimensionar el generador fotovoltaico y las protecciones.

### 7.2. Consumo diario y determinación del ángulo de inclinación

Como se ha calculado en el Capítulo 6, el consumo de energía diario es de 153.8 Wh, dado que el sistema consume unos 0.267A a 12 V de valor promedio durante las 24 horas del día y se ha aplicado un factor de seguridad de 2. Con este valor, se procede a aplicar los rendimientos del sistema en la ecuación 7.1. Considerando los valores recomendados en [47], se aplica un rendimiento del

	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Radiación solar diaria ( $kWh/m^2$ )	3.29	3.90	4.84	4.74	5.39	5.90	6.39	5.44	4.97	4.24	3.49	3.18

Tabla 7.1: Radiación solar diaria en Los Cristianos[26].

	28°	29°	30°	32°	34°	36°	38°	40°	42°	44°	46°
<b>Enero</b>	2974	2986	2998	3021	3040	3056	3070	3081	3090	3095	3098
<b>Febrero</b>	4848	4866	4884	4915	4941	4963	4979	4991	4998	5000	4997
<b>Marzo</b>	5223	5225	5226	5224	5217	5205	5188	5166	5139	5107	5070
<b>Abril</b>	4676	4660	4644	4608	4568	4524	4477	4425	4370	4311	4249
<b>Mayo</b>	4942	4912	4882	4819	4751	4679	4603	4524	4440	4353	4263
<b>Junio</b>	5191	5152	5113	5030	4943	4852	4756	4657	4553	4446	4335
<b>Julio</b>	5675	5635	5593	5505	5412	5314	5212	5105	4993	4876	4756
<b>Agosto</b>	5210	5187	5163	5111	5055	4994	4928	4858	4784	4706	4623
<b>Septiembre</b>	5262	5256	5250	5233	5211	5184	5152	5116	5075	5029	4978
<b>Octubre</b>	5092	5106	5119	5141	5159	5171	5178	5180	5177	5168	5155
<b>Noviembre</b>	4633	4661	4687	4736	4781	4821	4855	4885	4910	4930	4945
<b>Diciembre</b>	4570	4605	4639	4705	4765	4820	4871	4916	4956	4991	5021

Tabla 7.2: Irradiación solar diaria ( $Wh/m^2$ ) según inclinación.

98% para el sistema y del 85% para la batería, obteniendo el consumo diario mostrado en la ecuación 7.2.

$$L_{MD} = \frac{L_{MDCC}}{\eta_{sist} \cdot \eta_{bateria}} \quad (7.1)$$

$$L_{MD} = \frac{153,8}{0,98 \cdot 0,85} = 184,63 \text{ Wh} \quad (7.2)$$

Con un consumo de 184.63 Wh, se procede a estudiar la inclinación ideal de los paneles fotovoltaicos. Se ha elegido como posible ubicación, la zona cercana a la costa de los Cristianos (Latitud de 28° 03' 10.48"N y Longitud de 16° 43' 11.26" O). Se ha consultado la base de datos de GRAFCAN [26] y se ha extraído la radiación solar diaria de cada mes sobre superficie horizontal. Estos datos se muestran en la Tabla 7.1.

Posteriormente, se procede a calcular el valor de irradiación solar diaria a diferentes inclinaciones. Para ello, se hace uso de una herramienta de cálculo proporcionada por la Universidad de Jaen [38]. En ella se introducen los valores de radiación, la latitud (28° en Canarias), el acimut<sup>1</sup> (se ha considerado 0°) y la inclinación deseada. Se ha calculado para inclinaciones que parten de los 28° hasta los 46° en incrementos de 2°. Los resultados se muestran en la Tabla 7.2.

<sup>1</sup>Ángulo que forma el meridiano con el círculo vertical que pasa por un punto de la esfera celeste o del globo terráqueo.

	28°	29°	30°	32°	34°	36°	38°	40°	42°	44°	46°
<b>Enero</b>	<b>0,0311</b>	<b>0,0309</b>	<b>0,0308</b>	<b>0,0306</b>	<b>0,0304</b>	<b>0,0302</b>	<b>0,0301</b>	<b>0,0300</b>	<b>0,0299</b>	<b>0,0298</b>	<b>0,0298</b>
<b>Febrero</b>	0,0191	0,0190	0,0189	0,0188	0,0187	0,0186	0,0186	0,0185	0,0185	0,0185	0,0185
<b>Marzo</b>	0,0177	0,0177	0,0177	0,0177	0,0177	0,0177	0,0178	0,0179	0,0180	0,0181	0,0182
<b>Abril</b>	0,0198	0,0198	0,0199	0,0200	0,0202	0,0204	0,0206	0,0209	0,0211	0,0214	0,0217
<b>Mayo</b>	0,0187	0,0188	0,0189	0,0192	0,0194	0,0197	0,0201	0,0204	0,0208	0,0212	0,0217
<b>Junio</b>	0,0178	0,0179	0,0181	0,0184	0,0187	0,0190	0,0194	0,0198	0,0203	0,0208	0,0213
<b>Julio</b>	0,0163	0,0164	0,0165	0,0168	0,0171	0,0174	0,0177	0,0181	0,0185	0,0189	0,0194
<b>Agosto</b>	0,0177	0,0178	0,0179	0,0181	0,0183	0,0185	0,0187	0,0190	0,0193	0,0196	0,0200
<b>Septiembre</b>	0,0176	0,0176	0,0176	0,0177	0,0177	0,0178	0,0179	0,0181	0,0182	0,0184	0,0186
<b>Octubre</b>	0,0181	0,0181	0,0180	0,0180	0,0179	0,0179	0,0178	0,0178	0,0178	0,0179	0,0179
<b>Noviembre</b>	0,0199	0,0198	0,0197	0,0195	0,0193	0,0192	0,0190	0,0189	0,0188	0,0187	0,0187
<b>Diciembre</b>	0,0202	0,0201	0,0199	0,0196	0,0194	0,0192	0,0190	0,0188	0,0186	0,0185	0,0184

Tabla 7.3: Tabla con la relación consumo/irradiación para diferentes inclinaciones y meses.

Una vez obtenidos estos valores, se calcula la relación entre el consumo de energía diario y la irradiación y se selecciona el mayor de los valores de cada inclinación, y de estos el menor de todos como el mes crítico y el ángulo adecuado (Tabla 7.3). Se obtiene que la **inclinación adecuada** oscila entre los **44°** y **46°**, siendo el mes crítico Enero.

### 7.3. Dimensionado del generador fotovoltaico

El siguiente paso es calcular las Horas Solares de Pico (HSP). Para ello, se divide la irradiación solar diaria (ISD) entre  $1000\text{W}/\text{m}^2$ , como se muestra en la Ecuación 7.3. El valor obtenido es de 4.65 horas.

$$HSP = \frac{ISD}{1000\text{ W}/\text{m}^2} = \frac{4647\text{ Wh}}{1000\text{ W}/\text{m}^2} = 4,647\text{ h} \quad (7.3)$$

Con dicho valor, se calcula la energía a proporcionar por los paneles (Ecuación 7.4) como el producto de la potencia máxima del panel en condiciones STC<sup>2</sup> por las HSP y el *Performance Ratio* (PR). Se han considerado paneles de 60 W y un PR de 0.8 como recomiendan en [47].

$$E_{DMF} = P_{MOD,STC} \cdot HSP \cdot PR = 60 \cdot 4,647 \cdot 0,8 = 223,08\text{W} \quad (7.4)$$

Con dicho valor se calcula el número total de paneles como muestra la Ecuación 7.5. Se necesita, por tanto, un único panel de 60 W para abastecer el sistema.

<sup>2</sup>Las condiciones STC son las condiciones estándar de los ensayos de paneles fotovoltaicos. Se dan a una irradiancia de  $1000\text{ W}/\text{m}^2$ , una temperatura de 25°C y una masa de aire de 1,5.

Potencia MPP	60 W
Tensión nominal	12 V
Tensión de circuito abierto	22.7 V
Tensión MPP	18.3 V
Corriente de cortocircuito	3.48 A
Corriente MPP	3.28 A

Tabla 7.4: Características del panel elegido.

$$N_T = \frac{L_{MD}}{E_{DMF}} = \frac{184,63 \text{ Wh}}{223,08} = 0,827 \quad (7.5)$$

El panel seleccionado es un panel de silicio policristalino *SolarPower* de la marca *Xunzel* con tensión nominal 12V y potencia máxima de 60 W. Sus características técnicas se muestran en la Tabla 7.4.

## 7.4. Dimensionado del regulador de carga

Se procede a calcular la corriente de entrada al regulador de carga MPPT.

$$I_{ENTRADA} = 1,25 \cdot I_{MOD,SC} \cdot N_p = 1,25 \cdot 3,48A \cdot 1 = 4,35A \quad (7.6)$$

La tensión máxima que debe soportar el regulador se corresponde con la  $V_{oc}$  del generador, que en este caso es de 22.7 Voltios.

El regulador seleccionado es un regulador MPPT del fabricante EPEver, modelo EPSolar Tracer AN3210 (véase Anexo H.2).

## 7.5. Dimensionado de las protecciones

Los fusibles de protección a la salida del panel fotovoltaico y del regulador MPPT, serán de tensión al menos un 20% superior a la tensión de circuito abierto del módulo o conjunto de módulos[47]. En este caso, su voltaje es igual o superior a 27.24V.

Por otro lado, la intensidad nominal del fusible debe cumplir (para una zona climática moderada como Canarias) lo expuesto en la Ecuación 7.7 [47, pág 106], por lo que su valor debe ser igual o superior a 4.87 Amperios. Se emplearán fusibles gPV 10x38 con corriente nominal 6A.

$$I_N \geq 1,4 \cdot I_{SC,STC} = 1,4 \cdot 3,48A = 4,87A \quad (7.7)$$

En general, también debe tenerse en cuenta la corriente inversa que es susceptible de circular por un panel fotovoltaico [47, pág 107] y que se expresa como  $I_{INV} = (\text{numero modulos} - 1) \cdot 1,2 \cdot I_{SC,STC}$ . No obstante como en este caso no existen ramas en paralelo, no se contempla el uso de fusibles de protección de corriente inversa.

## 7.6. Simulación de los cálculos

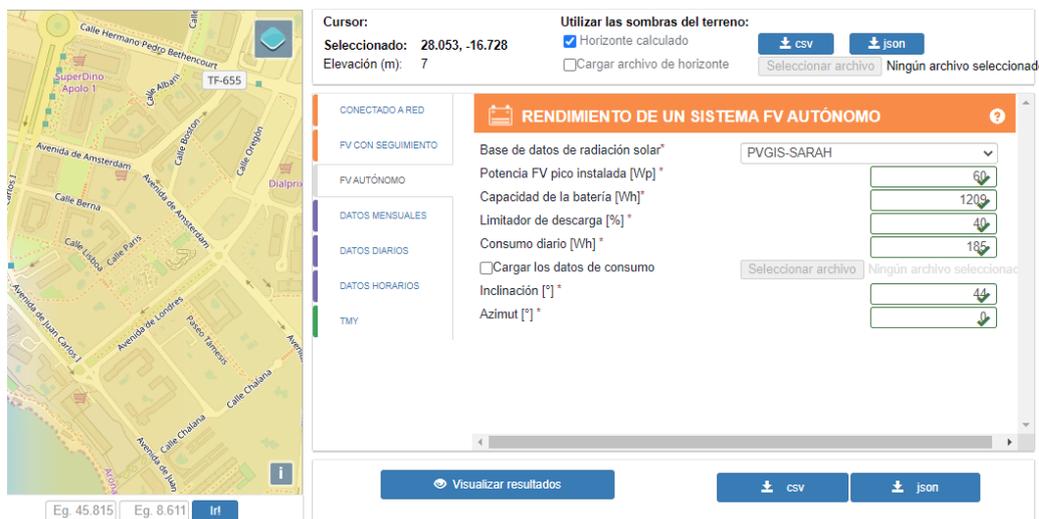


Figura 7.1: Captura de los datos introducidos en el simulador.

Con el fin de corroborar el correcto dimensionado del generador y la batería, se ha recurrido al uso de la herramienta de cálculo fotovoltaico proporcionada por PVGIS [17]. Se han introducido los datos de potencia del panel, consumo diario de energía, capacidad de la batería y ubicación, como se muestra en la captura de la Figura 7.1.

Tras realizar la simulación, se observa en la Figura 7.2 que tras finalizar el día, en el 59 % de los días la capacidad de la batería se mantiene superior al 92 %, el 24 % de los días entre el 84 % y 92 % y únicamente el 1 % de los días en un valor inferior al 30 % aunque superior al 20 %. Por tanto, se comprueba que en ningún momento el sistema se quedaría sin suministro. El resto del reporte de resultados puede ser visualizado en el Anexo G.

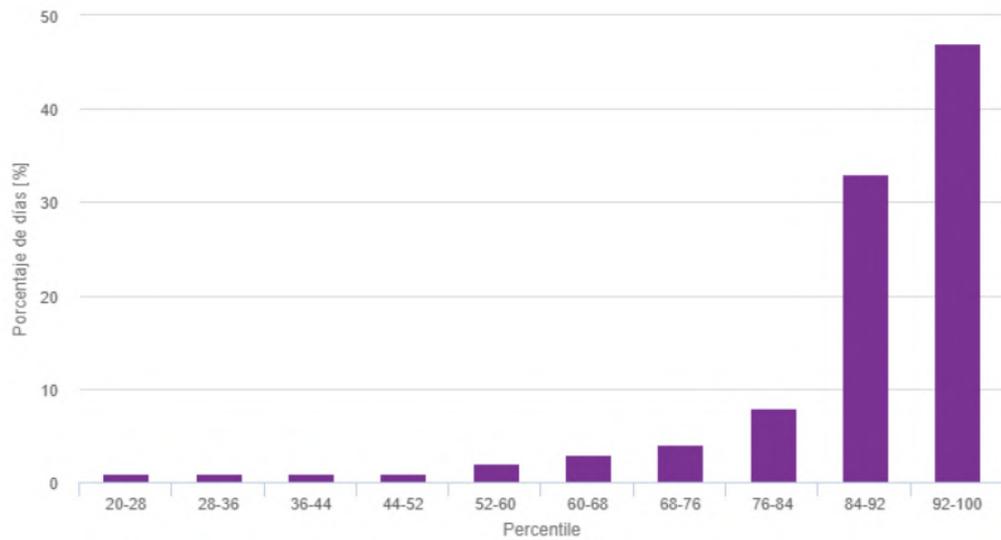


Figura 7.2: Probabilidad del SOC de la batería al finalizar el día (ver Anexo G.2).

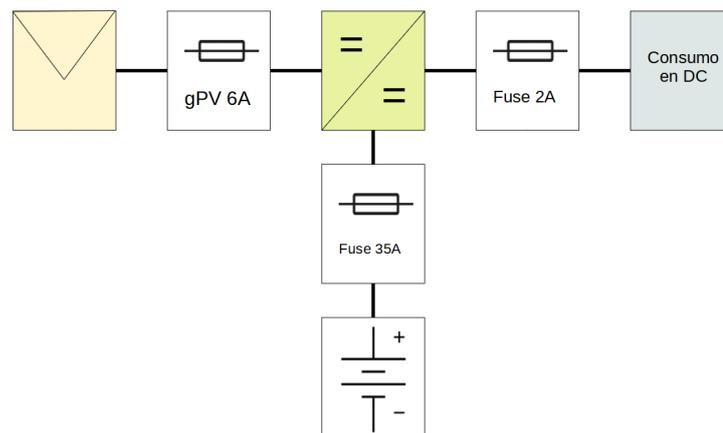


Figura 7.3: Esquema de instalación del sistema adoptado.

## 7.7. Esquema de instalación

Por último, en la Figura 7.3 se muestra el esquema de instalación adoptado. En él se puede observar como se emplean fusibles de protección a la entrada y salida del regulador MPPT, así como en la entrada de la propia batería. Por último, se observa que los consumos se han conectado al regulador de carga. De esta forma, se ahorra un relé en la instalación al permitir encender o apagar los consumos

controlando el relé integrado en el regulador haciendo uso de la comunicación CAN y el módulo **EPSolar-CAN**.



# Capítulo 8

## Desarrollo de *Software*

A lo largo del presente capítulo se detalla el *software* desarrollado en los microcontroladores de los módulos de BMS y CAN2Serial, así como la configuración del BUS CAN adoptada. Por otro lado, se explica también el *software* del dispositivo medidor de consumo, la interfaz gráfica de configuración y las herramientas de caracterización de las celdas. Todo el código implementado en este proyecto puede localizarse en el repositorio propio de GitHub [46].

### 8.1. Módulo BMS

Como se ha comentado en capítulos previos, el Sistema de Gestión de Baterías está formado por un microcontrolador que es programado mediante el lenguaje C++ y una serie de periféricos conectados por SPI como son el multiplexor, módulos de comunicación, etc. Para su programación se hace uso de algunas librerías propias del microcontrolador como la librería `SPI.h` o `EEPROM.h`. El resto de librerías empleadas, son de desarrollo propio, salvo la librería `LT_SPI.h` proporcionada por Analog Devices [45] y la `mcp2515.h` proporcionada por Travis a través de GitHub[24].

El esquema de ejecución cíclico del BMS puede observarse en el diagrama de la Figura 8.1. Una vez ejecutada la configuración en el arranque, de forma cíclica realiza en primer lugar la comunicación vía SPI con el módulo **LTC6804** para obtener las lecturas de los registros correspondientes al voltaje de las celdas. Tras esto, se convierte el voltaje de las celdas a un formato adecuado para ser enviado vía CAN BUS (se explica su codificación más adelante) y si ha pasado el tiempo configurado desde el envío del anterior mensaje, proceder a enviar por el bus los mensajes de voltaje de batería. Se envía de 1 a 3 mensajes según el número de celdas conectadas (cada mensaje contiene información de 4 celdas).

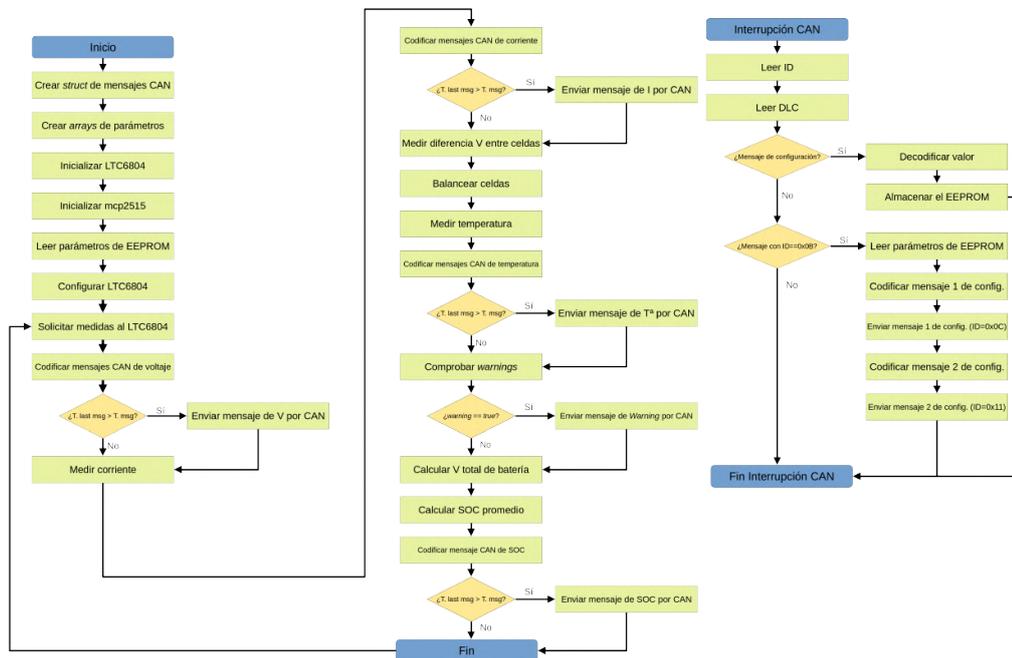


Figura 8.1: Diagrama de flujo del *firmware* del BMS (Anexo L.1).

Tras esto, se verifica si alguna de las celdas debe ser balanceada. En primer lugar, se comprueba que no se haya forzado el balanceo, dado que en ese caso existe otra rutina para la activación. Si el balanceo está desactivado en la configuración, se desactivan todos los transistores que controlan el balanceo. En caso de tener activo el balanceo en carga, en descarga o ambas, se comprueba primero dicha condición y tras eso, se comprueba celda a celda si la diferencia con la menor celda es superior al umbral establecido. En ese caso, se activa el transistor correspondiente al balanceo de dicha celda.

Posteriormente, se calcula el voltaje de la batería y en caso de existir una *sobretensión* o *subtensión* se envía un mensaje de advertencia por CAN. Tras esto, la siguiente etapa consiste en la lectura de las celdas. Para ello, se seleccionan las entradas del *MUX* según el número de termistores configurados. Se lee el valor analógico y se realiza su conversión a una temperatura, para posteriormente codificarlos de forma adecuada para enviarlo en un mensaje CAN.

Concluida esta etapa, se procede a calcular el estado de carga. Primero se mide la corriente, se codifica y se envía su valor por el BUS CAN y tras esto, con los parámetros de la batería (resistencia interna, número de celdas en serie, en paralelo, etc.) se calcula el SOC haciendo uso del método de la *Tensión de Circuito Abierto* y la curva de grado 4 explicada en el Capítulo 6. Finalmente, se

codifica y se envía por CAN.

Para la recepción de mensajes CAN se hace uso de interrupciones. Cuando un mensaje CAN ha pasado la máscara y el filtro, que posteriormente se explicará, se genera una interrupción que provoca que se ejecute la rutina `can_msg_rcv()`. En esta rutina, se almacena el mensaje CAN y se lee en primer lugar su identificador (ID). Según el valor del ID, se decodifica el mensaje de una determinada forma y se procede, en caso de ser un mensaje de configuración, a almacenarlo en la EEPROM y en caso contrario a ejecutar la acción indicada en el mensaje.

### 8.1.1. Comunicación con el LTC6804

Como se ha comentado anteriormente, para realizar la comunicación con el módulo de medición de celdas, se hace uso de la librería `LTC6804.h` y `LT_SPI.h` de Analog Devices. Estas librerías permiten acceder a los registros (de 1 byte) deseados del circuito integrado (CI) vía SPI a una frecuencia de 1 MHz.

El primer paso para la comunicación, consiste en la configuración del circuito integrado, indicando el número de canales de medición de celdas, la activación de los GPIO y si se permite o no el balanceo de las celdas durante la medición. Estos parámetros son denominados *CH*, *CHG* y *DCP*, respectivamente. Para ello, se hace uso de la función `set_adc()` que según las constantes que se le proporcionen realizan dichas configuraciones. Esta función es ejecutada dentro de la función `LTC6804_initialize()` donde se inicializa en primer lugar el bus SPI a la frecuencia antes indicada.

Para almacenar los mensajes de transmisión y recepción de configuración del LTC6804, se crean dos *arrays* de tipo byte (`uint8t`). El primero, destinado a almacenar los mensajes a transmitir al integrado, tiene como primera dimensión el número de circuitos integrados y como segunda 6. De esta forma, se crea un *array* de *n* filas y 6 columnas de 1 byte. Cada byte corresponde a uno de los registros CFGR que indica el *datasheet* y que se muestran en la Figura 8.2. El segundo, destinado a almacenar los mensajes de la configuración almacenada en el integrado, presenta como primera dimensión también el número de circuitos integrados y como segunda 8 bytes. Los 6 primeros bytes almacenan los mismos registros indicados en la Figura 8.2, y los 2 últimos el CRC para verificar la correcta recepción de los mensajes.

Se crea también, otro *array* de elementos de 16 bits con una distribución similar, aunque con una longitud de 12, para almacenar la información de los voltajes de las celdas medidas por el integrado, que se localizan en los registros CVAR0 a CVAR5, CVBR0-CVBR5, CVCR0-CVCR5 y CVDR0-CVDR5. A modo de ejemplo, se muestra la distribución de la medición de las tres primeras celdas en la Figura 8.3.

Tras finalizar la configuración del integrado, se procede a leer la tensión de las celdas, haciendo uso de la función `read_cell_voltage()` en la que se indica el

Table 36. Configuration Register Group

REGISTER	RD/WR	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CFGR0	RD/WR	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	REFON	SWTRD	ADCOPT
CFGR1	RD/WR	VUV[7]	VUV[6]	VUV[5]	VUV[4]	VUV[3]	VUV[2]	VUV[1]	VUV[0]
CFGR2	RD/WR	VOV[3]	VOV[2]	VOV[1]	VOV[0]	VUV[11]	VUV[10]	VUV[9]	VUV[8]
CFGR3	RD/WR	VOV[11]	VOV[10]	VOV[9]	VOV[8]	VOV[7]	VOV[6]	VOV[5]	VOV[4]
CFGR4	RD/WR	DCC8	DCC7	DCC6	DCC5	DCC4	DCC3	DCC2	DCC1
CFGR5	RD/WR	DCTO[3]	DCTO[2]	DCTO[1]	DCTO[0]	DCC12	DCC11	DCC10	DCC9

Figura 8.2: Tabla de los Registros de Configuración[45].

Table 37. Cell Voltage Register Group A

REGISTER	RD/WR	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CVAR0	RD	C1V[7]	C1V[6]	C1V[5]	C1V[4]	C1V[3]	C1V[2]	C1V[1]	C1V[0]
CVAR1	RD	C1V[15]	C1V[14]	C1V[13]	C1V[12]	C1V[11]	C1V[10]	C1V[9]	C1V[8]
CVAR2	RD	C2V[7]	C2V[6]	C2V[5]	C2V[4]	C2V[3]	C2V[2]	C2V[1]	C2V[0]
CVAR3	RD	C2V[15]	C2V[14]	C2V[13]	C2V[12]	C2V[11]	C2V[10]	C2V[9]	C2V[8]
CVAR4	RD	C3V[7]	C3V[6]	C3V[5]	C3V[4]	C3V[3]	C3V[2]	C3V[1]	C3V[0]
CVAR5	RD	C3V[15]	C3V[14]	C3V[13]	C3V[12]	C3V[11]	C3V[10]	C3V[9]	C3V[8]

Figura 8.3: Tabla de Registros de las Celdas C1 a C3[45].

número de CI, de celdas y se obtiene el *array* con los valores de las tensiones. En dicha función, se comienza inicializando la conversión ADC de los valores de las tensiones y, posteriormente, si la lectura de los registros se ha realizado correctamente (lo que se obtiene con la función `LTC6804_rdcv()`), se van almacenando los valores en el *array* `cell_codes[] []`, de lo contrario se almacena el valor 65555 que indicará un error en la lectura.

```

1 void read_cell_voltage (const uint8_t TOTAL_IC, const uint8_t
    TOTAL_CELL, uint8_t tx_cfg[][6], uint16_t cell_codes[][12]) {
2     start_cell_voltage_ADC(TOTAL_IC, tx_cfg);
3     int8_t error;
4     error = LTC6804_rdcv(0, TOTAL_IC, cell_codes);
5     if (error == -1) {
6         for (int i = 0; i < TOTAL_IC; i++) {
7             for (int j = 0; j < TOTAL_CELL; j++) {
8                 cell_codes[i][j] = 65555;
9             }
10        }
11    }
12 }

```

Listing 8.1: Función de lectura de celdas.

Para forzar balanceo de las celdas se llama a la función `force_balancing()` a la que se le pasan como argumentos el número de circuitos integrados, un valor binario de 8 bits y un booleano con el que indicar si se balancean las celdas de la 1 a la 8 (valor en `true`) o de la 9 a la 12 (valor en `false`). Este valor de 8 bits permite seleccionar con 1 las celdas que se balancean, siendo el LSB el correspondiente a

celda 1 o 9, respectivamente. A modo de ejemplo, si se deseara forzar el balanceo de las celdas 1,3 y 7, se debería emplear la función de la siguiente forma:

```
force_balancing(TOTAL_IC, tx_cfg, 0b01000101, true);
```

Esta función, escribe en los registros de configuración los 8 y 4 bits, respectivamente, de las celdas que deben balancearse, tal y como se muestra a continuación:

```
1 void force_balancing (const uint8_t TOTAL_IC, uint8_t tx_cfg
2   [][][6], uint8_t cells_to_balance , bool group){
3   if( group){
4     for (uint8_t i=0 ; i< TOTAL_IC; i++){
5       tx_cfg[i][4] = cells_to_balance ;
6     }
7   }
8   else{
9     for (uint8_t i=0 ; i< TOTAL_IC; i++){
10      tx_cfg[i][5] = ((tx_cfg[i][5])& 0b11110000) | (
11      cells_to_balance & 0b00001111);
12    }
13  }
```

Listing 8.2: Función para forzar balancear celdas.

Si por el contrario, se desea comprobar todas las celdas y balancear solo las necesarias, se emplea la función `balancing()` a la que se le proporciona: el número de integrados, el *array* de tensiones de las celdas, la máxima diferencia de voltaje entre celdas, el número de celdas y los umbrales superior e inferior. Esta función obtiene la menor tensión y comprueba si cada celda presenta una diferencia superior a este umbral. Se guarda en un entero sin signo, en cada bit, si dicha celda debe balancearse o no y se llama a la función `force_balancing()`, explicada anteriormente, para que se balancee dicha celda. En la medición de las celdas hay que tener en cuenta que los valores recibidos deben multiplicarse por  $100\mu V$ .

```
1 void balancing(const uint8_t TOTAL_IC, uint16_t cell_codes
2   [][][12], uint8_t tx_cfg [][][6], float max_difference , uint8_t
3   TOTAL_CELL, float VOV_THR, float VUV_THR ){
4   float V_min=cell_codes[0][0]*0.0001;
5   float V_max = cell_codes[0][0]*0.0001;
6   static uint16_t cell_to_balance = 0b0000000000000000;
7
8   //Se obtiene V_min y V_max
9   for (int i=0; i< TOTAL_IC; i++){
10    for (int j=0; j<TOTAL_CELL; j++){
11      if(float(cell_codes[i][j]* 0.0001) <= V_min){
12        V_min = float(cell_codes[i][j]* 0.0001);
13      }
14      if(float(cell_codes[i][j]* 0.0001) >= V_max){
15        V_max = float(cell_codes[i][j]* 0.0001);
16      }
17    }
18  }
```

```

14     }
15   }
16 }
17
18 //Se busca que celdas balancear
19 for (int i=0; i< TOTAL_IC; i++){
20   for(int j=0; j<TOTAL_CELL; j++){
21     float cell_volt = cell_codes[i][j]*0.0001;
22     if((((cell_volt - V_min) >= (max_difference))&(cell_volt >
VUV_THR))|| (cell_volt >= VOV_THR)){
23       cell_to_balance = (cell_to_balance | (0b1 << j));
24     }
25     else{
26       if((cell_volt - V_min) <= (max_difference -0.0003)&&(
cell_volt < (VOV_THR - 2*max_difference))){
27         cell_to_balance = (cell_to_balance & ~(uint16_t(0b1 <<
j)));
28       }
29     }
30   }
31 }
32
33 //Se balancean las celdas indicadas
34 uint8_t cell_to_bal_A = cell_to_balance & 0b0000000011111111;
35 uint8_t cell_to_bal_B = ((cell_to_balance >> 8) & 0
b0000000000001111);
36 force_balancing (TOTAL_IC, tx_cfg, cell_to_bal_A, true);
37 force_balancing (TOTAL_IC, tx_cfg, cell_to_bal_B, false);
38 LTC6804_initialize();
39 LTC6804_wrcfg(TOTAL_IC, tx_cfg);
40 }

```

Listing 8.3: Función para balancear celdas.

### 8.1.2. Lectura de temperaturas y corriente

Para la lectura de las temperaturas se hace uso de la entrada 0 del ADC del microcontrolador, a la que a través del multiplexor SPI se conectan las 32 entradas de temperatura. La función empleada se denomina `measure_all_temp()`. A esta función se le proporcionan como argumentos: un *float array* de temperaturas de 32 elementos, el número de termistores, los parámetros adecuados para la conversión del voltaje medido a un valor de temperatura, el pin de entrada analógica y el pin *CS* del multiplexor. Devuelve, además de las temperaturas almacenadas en el *array*, el valor máximo de temperatura medido.

A continuación se muestra el código implementado para dicha función:

```

1 float measure_all_temp(float array_temp[], const uint8_t N_NTC,
    const int BETTA, const int To, const int Ro, const int Raux,
    const int Vcc, const uint8_t AnalogInput, const uint8_t
    PIN_SYNC_MUX) {
2     float T_max = 0;
3     for (uint8_t i = 0; i < N_NTC ; i++) {
4         array_temp[i] = measure_temp(BETTA, To, Ro, Raux, Vcc, i,
            AnalogInput, PIN_SYNC_MUX);
5         if (array_temp[i] > T_max) {
6             T_max = array_temp[i];
7         }
8     }
9     return T_max;
10 }

```

Listing 8.4: Función para medición de temperaturas.

Como se observa, esta función llama a `measure_temp()`, en la que se elige el canal del SPI y se convierte la lectura analógica a un valor de temperatura, que se muestra a continuación:

```

1 float measure_temp(const int BETTA, const int To, const int Ro,
    const int Raux, const int Vcc, const uint8_t channel, const
    uint8_t AnalogInput, const uint8_t PIN_SYNC_MUX) {
2     select_channel_MUX(channel, PIN_SYNC_MUX);
3     float v_medida = (float(Vcc)/1023.0)*float(analogRead(
        AnalogInput));
4     float rntc = float(Raux)/((Vcc/v_medida)-1);
5     float t_medida = (float(BETTA) / (log(rntc / float(Ro)) + (
        float(BETTA) / float(To)))) - 273;
6     return t_medida;
7 }

```

Listing 8.5: Función de lectura de un termistor.

### Selección del canal del MUX

Para la selección del canal del multiplexor debe enviarse un mensaje de 8 bits con la estructura que muestra la Figura 8.4. En primer lugar se envían la selección del canal como 5 bits, seguido de un bit *don't care* y los bits de *ENABLE* y *CS*. Así, para seleccionar el canal 5, por ejemplo, la trama a enviar debe ser 00X00100.

Esta selección se hace mediante la función `select_channel_mux()`, que calcula el mensaje a enviar según el canal elegido, configura el bus SPI a 8MHz y finalmente envía el mensaje. El código se muestra a continuación:

```

1 void select_channel_MUX(const uint8_t channel, const uint8_t
    PIN_SYNC_MUX) {
2     if (channel <= 32) {

```

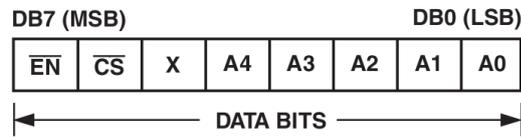


Figura 8.4: Trama del mensaje SPI del ADG731[2].

```

3  uint8_t msg = (0b00000000 | channel);
4  SPI.beginTransaction(SPISettings(8000000, MSBFIRST,
5  SPI_MODE2));
6  digitalWrite(PIN_SYNC_MUX, LOW);
7  SPI.transfer(msg);
8  digitalWrite(PIN_SYNC_MUX, HIGH);
9  SPI.endTransaction();
10 }

```

Listing 8.6: Función de selección del canal del multiplexor.

### Lectura de la corriente

La función `get_current()` permite medir la corriente que circula mediante el sensor **ACS712**. Para ello, toma el número de muestras indicado como parámetro y promedia todas las mediciones. Posteriormente, calcula el valor de corriente haciendo uso del parámetro de sensibilidad y devuelve el valor en mA, pudiendo añadirse un *offset*. Esta función se muestra a continuación:

```

1  int32_t get_current(const int samples_number, const float
2  sensibility_current, const uint8_t PIN_CURRENT_SENSOR, const
3  int16_t CURRENT_OFFSET){
4  float current_sum = 0;
5  for(int i=0; i<samples_number; i++){
6  float voltage = analogRead(PIN_CURRENT_SENSOR) * 5.0
7  /1023.0;
8  current_sum += (voltage -2.5)/sensibility_current;
9  }
10 int32_t current = round((current_sum/samples_number)*1000);
11 return(current + CURRENT_OFFSET);
12 }

```

Listing 8.7: Función para medir la corriente (en mA).

### 8.1.3. Almacenamiento de parámetros

Como se ha indicado anteriormente, el BMS es configurable. Por ello, los parámetros se deben almacenar en la memoria EEPROM interna que proporciona

el **ATMega 328P**. Se hace uso de un total de 12 registros de dicha memoria, almacenando cada parámetro en un byte.

Los parámetros de configuración que se almacenan se muestran en la Tabla 8.1, donde se indica además su modo de decodificación.

Dirección	Descripción	Decodificación
0x01	Configuración del CFGR0 (GPIO1-GPIO5 activo o inactivo)	$CFGRO = value$
0x02	Umbral de bajo voltaje de la celda (VUV)	$VUV = value \cdot 0,02 \text{ V}$
0x03	Umbral de sobre voltaje de la celda (VOV)	$VOV = value \cdot 0,02 \text{ V}$
0x04	Tiempo máximo que dura el balanceo	
0x05	Número de celdas en serie (NCELL)	$NCELL = value$
0x06	Número de termistores (NTEMP)	$NTEMP = value$
0x07	Tiempo de envío entre mensajes	$T\_SLEEP = value \cdot 15 \text{ ms}$
0x08	Diferencia máxima de voltaje entre celdas	$MAX\_DIFF = value \cdot 0,001 \text{ V}$
0x09	Tipo de balanceo	$BAL\_TYPE = value$
0x0A	Número de celdas en paralelo	$NCELL = value$
0x0B	Offset de corriente (mA) LSB	$OFFSET = ((MSB \ll 8)   LSB) - 32767 \text{ mA}$
0x0C	Offset de corriente (mA) MSB	$OFFSET = ((MSB \ll 8)   LSB) - 32767 \text{ mA}$

Tabla 8.1: Direcciones de memoria y decodificación de valores.

A estos parámetros se accede haciendo usos de dos funciones: `read_eeprom_ltc()` y `read_eeprom_atmega()`. La primera carga los valores de configuración del integrado **LTC6804** en el *array* de 8 bits `tx_cfg`, mientras que la segunda accede a los parámetros necesarios por el microcontrolador para su correcto funcionamiento.

A continuación, se muestra la función `read_eeprom_ltc()` implementada. Se lee el valor de voltaje mínimo (VUV), se convierte a su valor decimal, multiplicando por 0.02 y se procede a convertir a un valor codificado adecuadamente para el integrado. Para ello, se divide entre 0.0016 y se resta 1. Finalmente, este número de 12 bits se separa en su parte inferior de 8 bits, que se guarda en `tx_cfg[i][1]` y su parte superior que se almacena en `tx_cfg[i][2]`. Se procede igual para el valor de voltaje superior de la celda que se guarda en los 4 bits superiores de `tx_cfg[i][2]` y en `tx_cfg[i][3]`. Finalmente, el valor de DCTO se guarda en los 4 bits superiores del `tx_cfg[i][5]`.

```

1 void read_eeprom_ltc (const uint8_t TOTAL_IC, uint8_t tx_cfg
  [] [6]) {
2   uint16_t VUV = EEPROM.read(VUV_addr) * (0.02) * (1/0.0016) - 1;
3   uint16_t VOV = EEPROM.read(VOV_addr) * (0.02) * (1/0.0016) - 1;
4   for (uint8_t i=0 ; i < TOTAL_IC; i++){
5     tx_cfg[i][1] = VUV & (0b11111111);
6     tx_cfg[i][2] = (((VUV >> 8) & (0b00001111)) | ((VOV & 0
  b00001111) << 4));
7     tx_cfg[i][3] = ((VOV >> 4) & (0b11111111));
8     tx_cfg[i][5] = ((tx_cfg[i][5] & 0b00001111) | (EEPROM.read(
  DCTO_addr) << 4));

```

```

9 }
10 }

```

Listing 8.8: Función para leer parámetros de configuración del LCT6804 almacenados en la EEPROM.

Por otro lado, la función `read_eeprom_atmega()` almacena el resto de parámetros aplicando una metodología similar:

```

1 void read_eeprom_atmega(float &UV_THR, float &OV_THR, uint8_t &
  N_NTC, uint8_t &TOTAL_CELL, float &UVBAT_THR, float &
  OVBAT_THR, float &MAX_VCELL_DIFF, uint8_t &BALANCING_TYPE,
  uint8_t &NCELL_PARALLEL, int16_t &CURRENT_OFFSET, uint8_t &
  TSLEEP){
2  uint8_t value = uint8_t(EEPROM.read(NCELL_addr));
3  if (value <= 12){
4    TOTAL_CELL = value;
5  }
6  value = EEPROM.read(VUV_addr);
7  if (value <= 254){
8    UV_THR = float(EEPROM.read(VUV_addr))*0.02;
9  }
10  OV_THR = float(EEPROM.read(VOV_addr))*0.02;
11  value = uint8_t(EEPROM.read(N_NTC_addr));
12  if (value <= 32){
13    N_NTC = value;
14  }
15  UVBAT_THR= UV_THR * TOTAL_CELL;
16  OVBAT_THR = OV_THR * TOTAL_CELL;
17  MAX_VCELL_DIFF = float(EEPROM.read(MAX_DIFF_CELL_addr)*0.001);
18  value = uint8_t(EEPROM.read(BALANCING_TYPE_addr));
19  if(value <=3){
20    BALANCING_TYPE = value;
21  }
22  if (value <= 99){
23    NCELL_PARALLEL = value;
24  }
25  CURRENT_OFFSET = int16_t((uint16_t(uint8_t(EEPROM.read(
    CURRENT_OFFSET_addr2)) << 8) | uint16_t(EEPROM.read(
    CURRENT_OFFSET_addr))) - 32767);
26  value = uint8_t(EEPROM.read(TSLEEP_addr));
27  if(value <= 220){
28    TSLEEP = value;
29  }
30  else{
31    TSLEEP = 0;
32  }

```

33 }  
}

Listing 8.9: Función de lectura de parámetros de funcionamiento del BMS desde la EEPROM.

### 8.1.4. Estimación del SOC

Una vez leída la corriente de carga o descarga de la batería y la tensión de cada celda, se procede a estimar el estado de carga. Como se ha comentado anteriormente, se hace uso de la aproximación de la curva de SOC-Voc de grado 4. Para ello, se emplea una función llamada `calculate_SOC()` a la que se le debe proporcionar la corriente de descarga de la celda, la tensión medida y la resistencia interna de la celda. Para estimar el SOC se toma como aproximación que cada celda en paralelo proporciona una cantidad de corriente exactamente igual, por lo que se divide la corriente medida entre el número de celdas en paralelo. Como tensión de circuito abierto, se toma el valor promedio de la tensión de todas las celdas. De este modo, al llamar a la función, quedaría de la siguiente manera:

```
float SOC = calculate_SOC(float(current/NCELL_PARALLEL),
float(voltaje_total/TOTAL_CELL), internal_resistor);
```

Esta función calcula en primer lugar la tensión de circuito abierto de la celda, a partir del voltaje y corriente medida y la resistencia interna. Tras esto, se evalúa el SOC a partir del valor de  $V_{oc}$  antes calculado. Es importante, tener en cuenta el criterio de signos: Se considera positiva la corriente de carga y negativa la de descarga. A continuación, se muestra el código de dicha función:

```
1 float calculate_SOC(int current, float voltage, float resistor)
2 {
3     float Voc = voltage - float(current)/1000.0 * resistor;
4     float SOC = (-0.4203*pow(Voc, 4) + 5.6962*pow(Voc,3) -
5         28.2351*pow(Voc,2) + 61.1863*Voc - 49.1338)*100.0;
6     if (SOC > 100.0){
7         SOC = 100;
8     }
9     else if (SOC <= 0.0){
10        SOC = 0.0;
11    }
12    return SOC;
13 }
```

Listing 8.10: Función para calcular el Estado de Carga de la batería.

### 8.1.5. Comunicación CAN BUS

Para la comunicación CAN BUS se hace uso de la librería `mcp2515.h`. Esta librería, que se puede obtener directamente en GitHub [24], permite la comunicación de cualquier microcontrolador AVR con el circuito integrado MCP2515 a través del bus SPI. Admite la decodificación de tramas estándar o extendidas, así como 2 máscaras y 5 filtros de recepción y velocidades que oscilan desde los 10 kbps hasta 1 Mbps. En este proyecto la velocidad establecida es de 125 kbps, suficiente para el número de mensajes que se transmiten por el bus.

Las principales funciones de la librería que se han empleado han sido las de configuración: `setConfigMode()`, `setFilterMask()` y `setNormalMode()`, con las que se habilita el modo de configuración, se establece el valor binario de la máscara y se activa el modo de funcionamiento normal, respectivamente. Por otro lado, para el envío de los mensajes CAN se hace uso de la función `sendMessage()`. Adicionalmente, se han creado otras funciones específicas para la codificación de cada mensaje CAN y la inicialización del módulo. Estas pueden consultarse en el código implementado en [46].

Los mensajes CAN son almacenados en un `struct` de tipo `can_frame` definido en la propia librería[24]. Está formado por un campo para el ID (un entero sin signo de 32 bits), un entero de 8 bits para la longitud del mensaje (`can_dlc`) y un `array` de 8 bytes para los datos.

#### Mensajes CAN de configuración

A continuación se detallan los mensajes CAN que son transmitidos para configurar el BMS, indicando su identificador, su longitud y su codificación. Estos mensajes indican los valores que deben guardarse en las direcciones de memoria de la Tabla 8.1.

#### Configurar valor VUV

Este mensaje configura el valor de tensión mínima de la celda en el BMS. En la Figura 8.5 se muestra la estructura de dicho mensaje y su forma de decodificación. Este valor recibido se almacena en la dirección de memoria 0x02 de la EEPROM (`VUV_addr`).



Figura 8.5: Trama del mensaje CAN de voltaje mínimo.

### Configurar valor VOV

Este mensaje configura el valor de tensión máxima de la celda en el BMS, cuyo valor se almacena en la dirección de memoria 0x03 de la EEPROM (VOV\_addr). En la Figura 8.6 se muestra la estructura de dicho mensaje y su forma de decodificación.



Figura 8.6: Trama del mensaje CAN de voltaje máximo.

### Configurar valor DCTO

Este mensaje configura el tiempo que dura la descarga de la celda  $C_x$  durante el balanceo (en minutos). Este tiempo es configurable según el valor de data[0] en el mensaje CAN (Figura 8.7) según se muestra en la siguiente tabla.



Figura 8.7: Trama del mensaje CAN de DCTO.

### Configurar valor N\_CELL

Este mensaje (Figura 8.8) configura el número de celdas conectadas en serie en el BMS (un valor entre 1 y 12). Se toma dicha configuración a partir de la conversión directa del valor del mensaje a un entero de 8 bits sin signo y se almacena directamente en la dirección de memoria 0x05 de la EEPROM (NCELL\_addr).



Figura 8.8: Trama del mensaje CAN de configuración de celdas en serie.

### Configurar valor N\_CELL\_PARALLEL

Este mensaje (Figura 8.9) configura el número de celdas conectadas en paralelo en el BMS. Esto es un valor entre 1 y 99 que se toma de la conversión directa a un entero sin signo del byte recibido en el mensaje CAN. Dicho valor se almacena en la dirección de memoria 0x0A de la EEPROM (NCELL\_PARALLEL\_addr).

data[0]	Timeout $C_x$
0	Desactivado
1	0.5 minutos
2	1 minuto
3	2 minutos
4	3 minutos
5	5 minutos
6	5 minutos
7	10 minutos
8	15 minutos
9	20 minutos
10	30 minutos
11	40 minutos
12	60 minutos
13	75 minutos
14	90 minutos
15	120 minutos

Tabla 8.2: Tabla de conversión de tiempos de DCTO.



Figura 8.9: Trama del mensaje CAN de configuración de celdas en paralelo.

### Configurar valor $N_{NTC}$

Este mensaje configura el número de termistores conectados al BMS. Puede verse la trama CAN enviada en la Figura 8.10. El valor (un número entre 1 y 32) se obtiene de la conversión directa del byte recibido a un entero de 8 bits sin signo. Este valor se almacena en la dirección de memoria 0x06 de la EEPROM ( $N_{NTC\_addr}$ ).

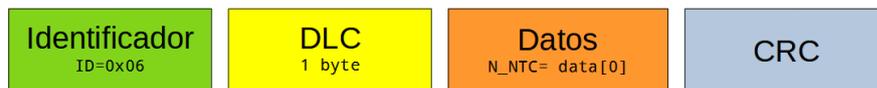


Figura 8.10: Trama del mensaje CAN de configuración del número de termistores.

### Configurar Tiempo entre Mensajes

Este mensaje configura el tiempo en milisegundos entre el envío de mensajes CAN. Su valor se obtiene del escalado del entero de un byte recibido por 100 ms. Este valor se almacena en la dirección de memoria 0x07 de la EEPROM (TSLEEP\_addr). Puede observarse esta distribución en la Figura 8.11.



Figura 8.11: Trama del mensaje CAN de configuración del tiempo entre mensajes.

### Forzar el balanceo de las Celdas 1 a la 8

Este mensaje (mostrado en la Figura 8.12) activa el balanceo de las celdas cuyo bit en el byte esté a 1, durante el tiempo almacenado en el DCT0. El bit menos significativo representa la celda 1, mientras el MSB representa la celda 8.



Figura 8.12: Trama del mensaje CAN para forzar el balanceo de las Celdas 1 a 8.

### Forzar el balanceo de las Celdas 9 a la 12

Este mensaje (Figura 8.13) activa el balanceo de las celdas 9 a 12 restantes. En este caso, el LSB corresponde a la celda 9 y el bit3 a la celda 12. Los bits desde el 4 hasta el 7 son ignorados.

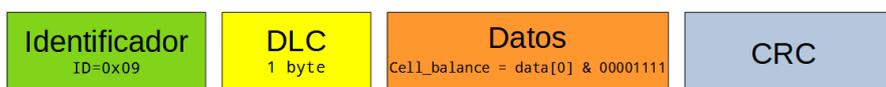


Figura 8.13: Trama del mensaje CAN para forzar el balanceo de las Celdas 9 a 12.

### Configurar el tipo de balanceo a realizar

Este mensaje configura si el balanceo de las celdas se realiza durante la carga, la descarga, en ambas o ninguna. Para ello, según el valor del mensaje de la Figura 8.14, se configura un tipo de balanceo de acuerdo a la Tabla 8.3. Este valor se almacena en la dirección de memoria 0x07 de la EEPROM (TSLEEP\_addr).



Figura 8.14: Trama del mensaje CAN para configurar cuando realizar el balanceo.

data[0]	Tipo de balanceo
0b00	Desactivado
0b01	Solo en la carga
0b10	Solo en la descarga
0b11	En carga y descarga

Tabla 8.3: Tabla con valores de configuración del tipo de balanceo.

### Configurar máxima diferencia entre celdas en mV

Este mensaje CAN (Figura 8.15) configura la máxima diferencia de tensión en mV admisible entre las diferentes celdas. El valor se reconstruye multiplicando el byte recibido (como un entero sin signo) por 1mV. Dicho valor se almacena en la dirección de memoria 0x08 de la EEPROM (MAX\_DIFF\_CELL\_addr).

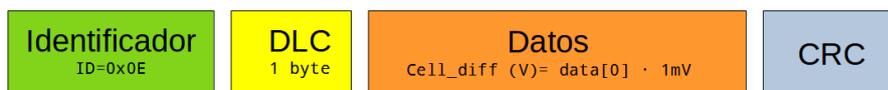


Figura 8.15: Trama del mensaje CAN para configurar la máxima diferencia entre celdas.

### Configurar *offset* del sensor de corriente (en mA)

Este mensaje CAN (Figura 8.16) configura el *offset* a aplicar en la medida del sensor de corriente. Permite un *offset* entre -32767 y +32767 mA. El valor (en mA) se reconstruye a partir de la conversión a un entero de 16 bits de los dos mensajes recibidos y posteriormente restarle 32767. Este se almacena en las

direcciones de memoria 0x0B y 0x0C de la EEPROM (CURRENT\_OFFSET\_addr y CURRENT\_OFFSET\_addr2).

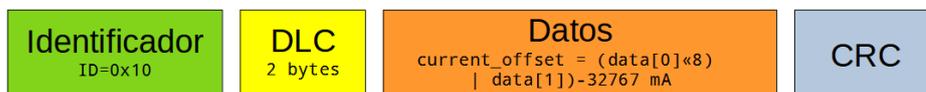


Figura 8.16: Trama del mensaje CAN para configurar el offset del sensor de corriente.

### Pedir las configuraciones del BMS

Este mensaje CAN pide todas las configuraciones almacenadas en la EEPROM del BMS (Figura 8.17). El BMS responde con un mensaje con  $ID = 0x0C$  y 8 bytes con los datos que ha obtenido de las direcciones de la EEPROM 0x00 a 0x07 y otro mensaje con  $ID = 0x11$  con 5 bytes con los datos obtenidos en las direcciones 0x08 a 0x0C.



Figura 8.17: Trama del mensaje CAN para pedir las configuraciones del BMS.

### Mensajes CAN de información (TPDO)

A continuación se detallan los mensajes CAN que son transmitidos con la información medida por el BMS, indicando su identificador, su longitud y su codificación. Estos mensajes son enviados de forma periódica según el tiempo configurado y almacenado en la dirección de memoria 0x07.

### Enviar todas las configuraciones almacenadas en el BMS

Una vez recibido el mensaje de pedir configuraciones con  $ID = 0x0B$ , el BMS responde con los dos mensajes que se detallan a continuación en la Figura 8.18. Estos mensajes están codificados de la misma forma que los mensajes de configuración explicados en el apartado anterior.

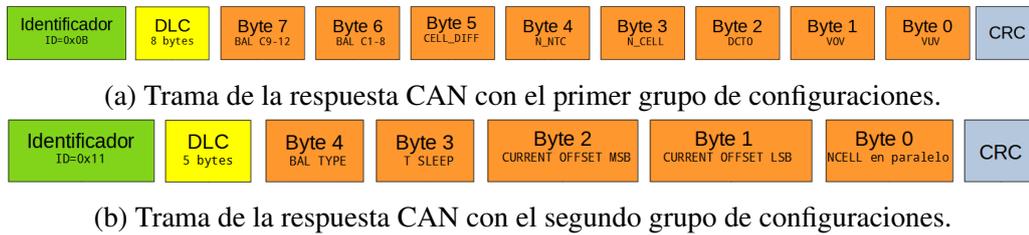


Figura 8.18: Mensajes CAN de respuesta con la configuración del BMS.

### Enviar los valores en Voltios de las celdas 1 a la 4

Este mensaje envía el voltaje de las celdas C1-C4 cada una de ellas como 2 bytes. El valor obtenido debe multiplicarse por  $100 \mu V$  para tener la tensión de la celda. Se procede de igual forma para las celdas C5-C8, empleando el  $ID = 0x41$  y las celdas C9-C12 con el  $ID = 0x42$  (Figura 8.19).

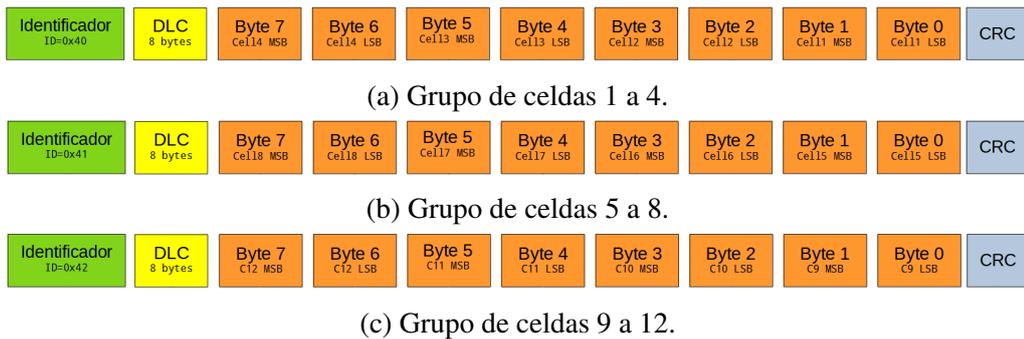


Figura 8.19: Mensajes CAN con las tensiones de las celdas.

A modo de ejemplo, a continuación se muestra la decodificación del valor de la *celda 1*, siendo el mismo procedimiento para el resto:

$$\text{Valor C1} = (\text{data}[0] \mid (\text{data}[1] \ll 8)) \cdot 100\mu V \quad (8.1)$$

### Enviar las temperaturas de las NTC

Se definen 4 mensajes con  $ID = 0x43, 0x44, 0x45$  y  $0x46$  (Figura 8.20), dedicado cada uno de ellos a un grupo de 8 termistores (NTC 0-7, 8-15, 16-23 y 24-32, respectivamente). Cada byte almacena una temperatura, que debe decodificarse restando 5 y multiplicando por 0.3, es decir:

$$\text{Temperatura} = (\text{data}[i] - 5) * 0.3$$

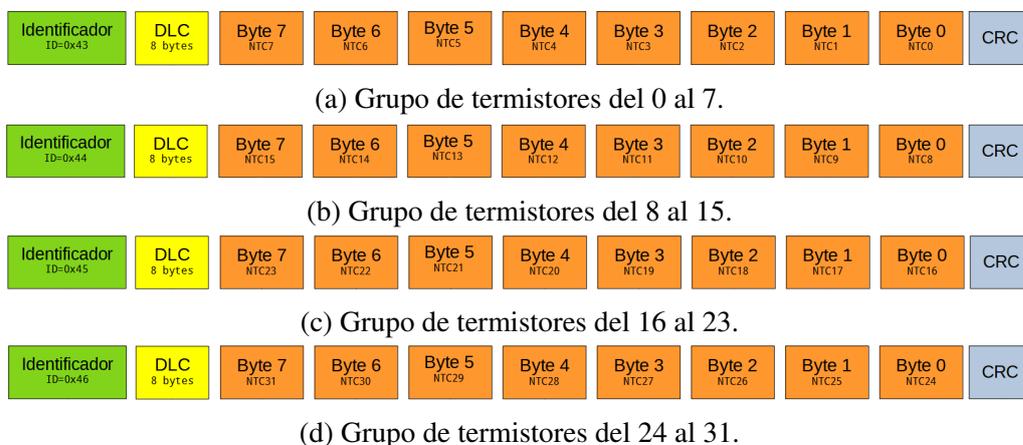


Figura 8.20: Mensajes CAN con las temperaturas medidas.

### Enviar el valor del SOC

Este mensaje (Figura 8.21) envía el SOC total de la batería como un valor entre 0-10000 (2 byte) correspondiente a 0-100% multiplicado por 100, que se decodifica como se muestra a continuación:

$$\text{Valor} = (\text{uint16\_t}(\text{data}[1] \ll 8) | \text{data}[0]) / 100.0 \quad (8.2)$$



Figura 8.21: Trama del mensaje CAN con el Estado de Carga.

### Enviar el valor del SOH

Este mensaje (Figura 8.22) envía el SOH de la batería como un valor entre 0-10000 (2 byte) correspondiente a 0-100% multiplicado por 100. Se decodifica de la misma forma que el mensaje de SOC.



Figura 8.22: Trama del mensaje CAN con el Estado de Salud.

### Enviar la corriente que circula por la batería

Este mensaje envía la corriente medida por el BMS (siendo positiva para la carga de la batería) y negativa para la descarga (Figura 8.23). Se envía el valor como un entero de 4 bytes con el valor de corriente en mA, que se decodifica como se muestra a continuación:

$$\text{Valor} = (\text{data}[3] \ll 24 \mid \text{data}[2] \ll 16 \mid \text{data}[1] \ll 8 \mid \text{data}[0]) \quad (8.3)$$



Figura 8.23: Trama del mensaje CAN con la corriente de la batería.

### Distribución de los mensajes CAN para varios dispositivos

Como se ha explicado anteriormente, los identificadores de los mensajes CAN de información, están en el rango 0x40 a 0x49, y los de configuración están en el rango 0x02 a 0x11. Estos últimos permanecerán constantes independientemente del número de BMS conectados al bus, sin embargo los mensajes de información deben variar su identificador según del BMS del que provengan. Por ello, el primero empleará el rango antes descrito, mientras que en caso de existir un segundo BMS, empleará el rango de IDs desde 0x50 a 0x59, continuando así hasta alcanzar los IDs 0xF9, permitiendo conectar de esta forma hasta 12 dispositivos al mismo bus.

## 8.2. Módulo CAN2Serial

El módulo CAN2Serial es el encargado de permitir establecer una conexión entre el BMS y el PC a través del CAN. Este dispositivo consta, por un lado de la interfaz CAN bus (con el mismo conjunto de integrados **MCP2515** y **TJA1050**) y por otro del USB, para permitir una comunicación serial con el PC.

Este módulo está continuamente leyendo del puerto serie, en busca de alguno de los mensajes de envío de configuración desde el PC. Por otro lado, cuando se recibe un mensaje CAN adecuado, se produce una interrupción que hace ejecutar la rutina de clasificación del mensaje llamada `can_msg_rcv()`. De esta forma, el bucle principal de ejecución del microcontrolador está formado por las funciones `serial_event()` que lee del puerto serial para almacenar lo leído en un String,

get\_data() que convierte el String leído con el valor de configuración en un entero, la función can\_msg\_rcv() que devuelve por serial el mensaje del CAN recibido, y finalmente una sentencia de switch() que según el valor del ID del mensaje serial recibido, envía el mensaje CAN adecuado. Este esquema de ejecución, puede verse con mayor detalle en la Figura 8.24.

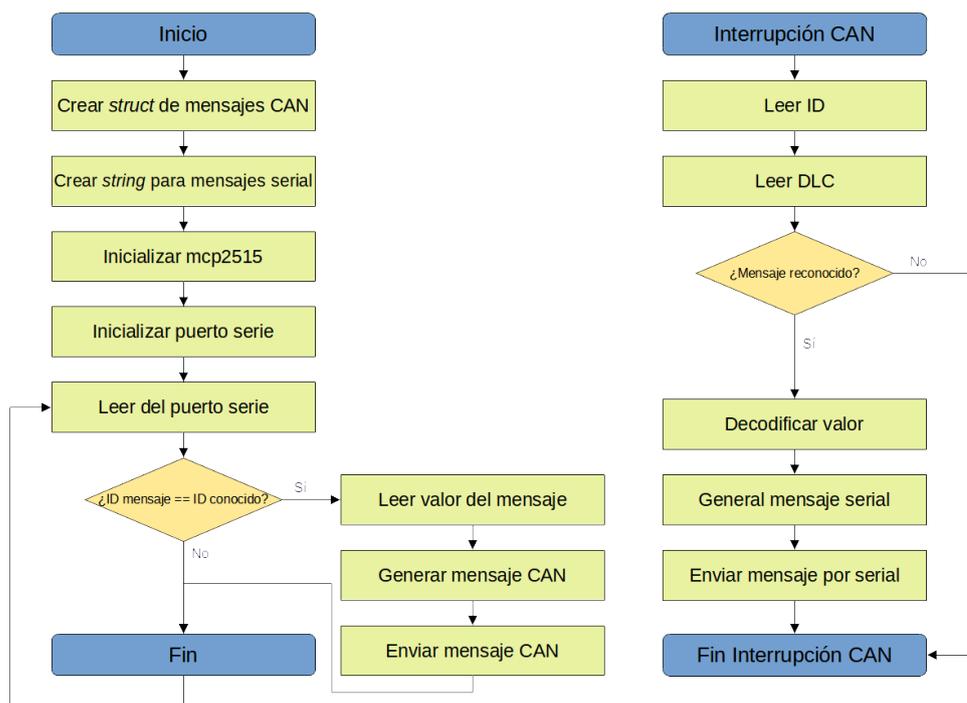


Figura 8.24: Flujograma del *firmware* del módulo CAN2Serial (Anexo L.2).

### 8.2.1. Mensajes Serial a enviar

A continuación, se detallan los diferentes mensajes que son enviados por puerto serie, por un lado para configurar el BMS y por otro, para lectura de parámetros, así como su codificación.

#### Mensajes serial de configuración

En este apartado, se muestran los mensajes que se deben enviar por puerto serie para configurar el BMS, estos mensajes son convertidos a mensajes CAN con el formato indicado en la sección 8.1.5. La configuración adoptada es 115200 baudios, con 8 bits de datos, 1 bit de parada y ningún bit de paridad. Se finaliza cada mensaje con un salto de línea '\n'.

### Configurar el valor de tensión mínima de celda (*VUV*)

Para configurar la tensión mínima, debe enviarse por serial el carácter ASCII 'A' seguido de un número entre 0 y 255. El valor en voltios corresponde al producto de dicho número por 0.02. Por ejemplo, si se envía el mensaje "A125", corresponde a una tensión de  $125 \cdot 0,02 = 2,5 \text{ V}$ .

### Configurar el valor de tensión máxima de celda (*VOV*)

Para configurar esta tensión, de forma similar al caso anterior, debe enviarse por serial el carácter ASCII 'B' seguido de un número entre 0 y 255 que será multiplicado por 0.02 para obtener el valor en voltios.

### Valor de *DCTO*

Para configurar el tiempo que dura activo el balanceo de una celda, debe enviarse por puerto serie desde el ordenador un mensaje encabezado por el carácter ASCII 'C', seguido de un valor entre 0 y 15, que configura el *timeout* como se muestra en la Tabla 8.4.

Valor de <i>DCTO</i>	<i>Timeout C<sub>x</sub></i>
0	Desactivado
1	0.5 minutos
2	1 minuto
3	2 minutos
4	3 minutos
5	5 minutos
6	5 minutos
7	10 minutos
8	15 minutos
9	20 minutos
10	30 minutos
11	40 minutos
12	60 minutos
13	75 minutos
14	90 minutos
15	120 minutos

Tabla 8.4: Tabla de conversión de tiempos de *DCTO*.

**Número de celdas conectadas**

Para configurar el número de celdas conectadas en serie, debe enviarse un mensaje encabezado por el carácter 'D' seguido de un número entre 1 y 12. Para configurar las celdas en paralelo, se envía un mensaje 'K' seguido de un número entre 1 y 99. Así, por ejemplo para configurar el BMS con 12 celdas en serie y 21 celdas en paralelo, debe enviarse por puerto serie los siguientes mensajes:

- "D12\n"
- "K21\n"

**Número de termistores conectados**

El número de termistores conectados se configura enviando un mensaje serial encabezado por 'E' y el número de sensores entre 1 y 32. Por ejemplo, para configurar 13 sensores conectados se envía el mensaje "E13\n".

**Máxima diferencia de voltaje entre celdas.**

La máxima diferencia admisible (en *mV*) de las celdas se configura con un mensaje encabezado por el carácter ASCII 'I' seguido de un valor entre 0 y 255.

**Activar el balanceo automático**

Es posible configurar diferentes casos para el balanceo de celdas: solo durante la carga, la descarga, en ambos o nunca. Para ello, se emplea el carácter 'J' seguido de un valor entre 0 y 3 según lo mostrado en la Tabla 8.5.

Mensaje serial	Tipo de balanceo
"J0\n"	Desactivado
"J1\n"	Solo en la carga
"J2\n"	Solo en la descarga
"J3\n"	En carga y descarga

Tabla 8.5: Tabla con valores de configuración del tipo de balanceo.

**Offset del sensor de corriente**

Es posible establecer el offset al sensor de corriente enviando el carácter ASCII 'L' seguido del valor numérico en mA. Por ejemplo, para un *offset* de -110 mA, se debe enviar el mensaje "L-110\n".

### Tiempo entre mensajes CAN

El tiempo (en ms) que debe pasar entre un mensaje CAN y el siguiente, puede configurarse enviando un valor entre 0 y 255, encabezado con el caracter ASCII 'F'. Por ejemplo, para un tiempo de 10 ms se envía el mensaje serial "F10\n".

### Forzar el balanceo de las celdas

Para activar el balanceo de las celdas, se dispone de dos mensajes diferentes: el primero, con el encabezado 'G' para las celdas de la 1 a la 8 y el segundo, con el encabezado 'H' para las celdas 9 a 12. Cada celda puede forzarse poniendo su bit a 1, enviándose el valor final en decimal. Por ejemplo, si se desea balancear las celdas 1, 4, 5 y 6 debe enviarse el mensaje "G57\n" (que en binario es 0b00111001).

### Pedir los valores de configuración del BMS

Existe un comando serial, con el que es posible solicitar todas las configuraciones almacenadas en el BMS. Este mensaje es "Z1\n". Una vez recibidas las configuraciones del BMS, estas se van devolviendo vía serial empleando los mismos identificadores y codificación que los mensajes de configuración explicados anteriormente ('A' para *VUV*, 'B' para *VOV*, 'C' para *DCTO*, etc.).

### Mensajes enviados por el BMS

El módulo CAN2Serial permite enviar por serial los mensajes CAN recibidos. Estos son: tensión de las celdas, temperatura de las NTC, valor de la corriente medida, valor de SOC y valor de SOH. A continuación, se detallan estos mensajes.

### Voltaje de las celdas

El voltaje de las celdas es enviado en tres mensajes diferentes: uno para cada grupo de tres celdas. Los identificadores son 'M', 'N' y 'O', para las celdas C1-C4, C5-C8, C9-C12, respectivamente. Cada mensaje serial, lleva el caracter correspondiente seguido de un número entre 1 y 4 para indicar la celda que corresponde y el valor del voltaje compuesto por 5 caracteres. Este valor debe ser multiplicado por 100  $\mu V$ . A continuación, se muestran dos ejemplos de mensajes de recepción y su decodificación.

- "M136000\n": Corresponde a la celda 1 del grupo 1 (C1) y su tensión es de  $36000 \cdot 100 \mu V = 3,6 V$ .

- "0337500\n": Corresponde a la celda 3 del grupo 3 (C11) y su tensión es de  $37500 \cdot 100 \mu V = 3,75 V$ .

### Temperatura medida por los termistores

Para el envío de las temperaturas se emplea el caracter ASCII 'P' seguido de 2 caracteres numéricos que indican el termistor (entre 1 y 32) y 3 caracteres numéricos que indican la temperatura. Para decodificar el valor de temperatura debe restarse 5 y multiplicarse por 0.3. Así, si por ejemplo se recibe el mensaje "P08225\n", significa que se está recibiendo la temperatura del termistor número 8 y su valor medido es  $66^{\circ}C ((225 - 5) \cdot 0,3 = 66^{\circ}C)$ .

### Valor de la corriente medida

El valor de la corriente medida (en mA) es enviada empleando el caracter 'Q', seguido del signo y de 5 caracteres numéricos con el valor. Así, si por ejemplo, la corriente medida es de 16.54 Amperios (en descarga), el mensaje que se envía por serial es "Q-16540\n".

### Valor del SOC y SOH

Para transmitir por serie los valores del estado de carga y el estado de salud, se emplean los caracteres 'T' y 'U', respectivamente, seguido de 4 caracteres numéricos que representan el porcentaje (multiplicado por 100). Si, por ejemplo, se recibe un mensaje CAN de un SOC del 97.68 %, el mensaje serial que se envía es "T9768\n".

## 8.2.2. Interfaz gráfica de configuración

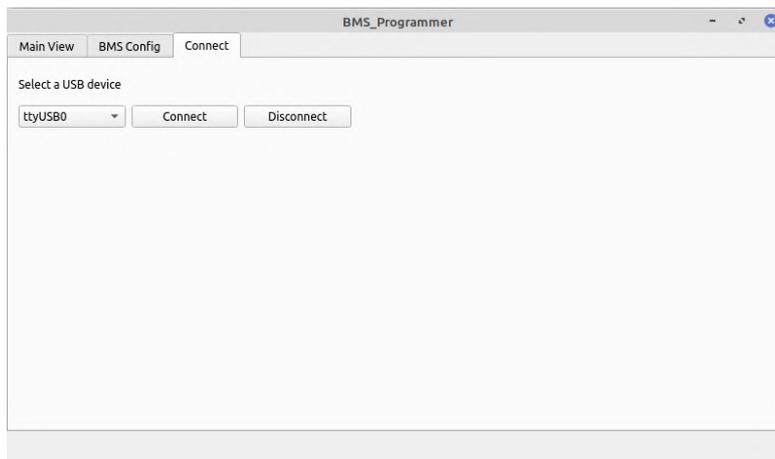
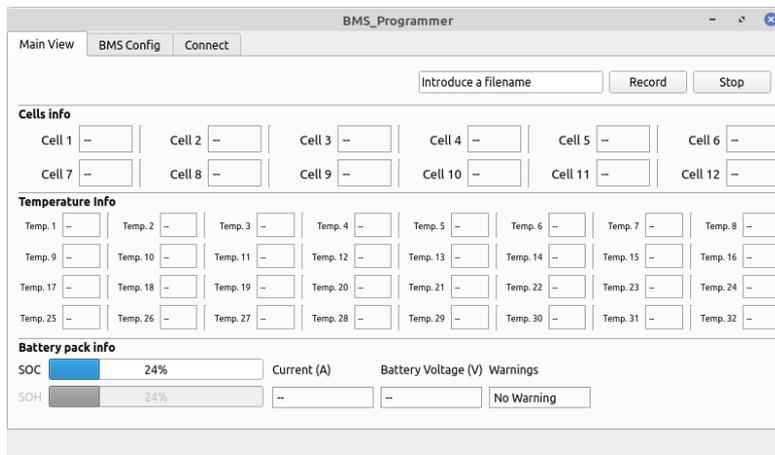
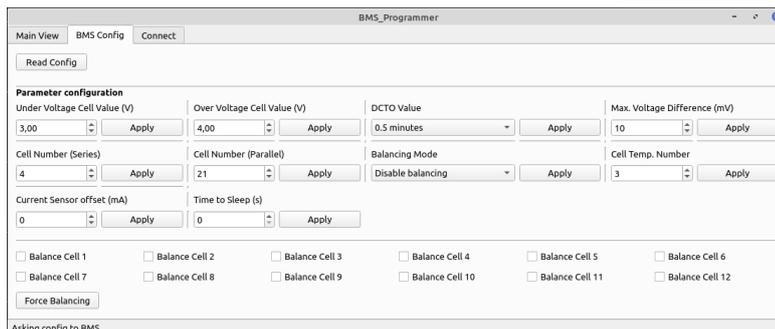
Para hacer posible la configuración vía CAN del BMS a través del CAN2Serial, se ha diseñado un *software* con interfaz gráfica, al que se ha denominado *BMS Programmer GUI*. Esta herramienta desarrollada en Python con la ayuda de PyQt, dispone de tres pestañas denominadas *Connect*, *BMS Config* y *Main View*, así como un *status bar* en la parte inferior que informa del estado del programa. La primera pestaña (que puede verse en la Figura 8.25a) permite seleccionar el puerto serie al que se ha conectado el módulo CAN2Serial. Una vez presionado el botón *Connect*, se comienza la ejecución de un *hilo* que lee de forma cíclica el puerto serial y va actualizando los valores que se observan en la pestaña *Main View*. En este mismo *hilo* de ejecución, cada 5 segundos se envía un comando "Z1\n" para actualizar las configuraciones. De esta forma, puede conocerse todas

las configuraciones almacenadas así como las celdas que están balanceándose en ese instante.

La pestaña *Main View* (Figura 8.25b) presenta tres grupos de datos. El primero de ellos corresponde a la información de las celdas. En ella puede verse la tensión de todas las celdas conectadas, mostrándose en rojo aquellas que se están balanceando. El segundo grupo corresponde a la información de temperatura, donde se muestra la temperatura de los termistores que están activos en ese instante. El tercer bloque, muestra información del estado de carga de la batería (con una barra deslizante), la corriente que circula por la batería (en Amperios), el voltaje total de la batería y los posibles *warnings* detectados.

En esta pestaña es posible crear un fichero *.csv* en el que se guardan los datos recibidos cada 10 segundos. Este fichero almacena la fecha y hora, el número de celdas en serie, el número de termistores, la corriente, tensión de las celdas, las celdas que se están balanceando, temperaturas, SOC, SOH y la tensión total de la batería. Esta rutina se realiza mediante la creación de un nuevo hilo que se dedica a la apertura del fichero, su actualización y posterior cierre, para evitar así la pérdidas de datos en caso de error del programa.

Por último, la pestaña *BMS Config* (Figura 8.25c) permite visualizar y editar los parámetros del BMS. Para ello, es preciso emplear los cursores para editar el valor, y posteriormente con el botón *Apply*, proceder a enviar dicho valor al BMS. Se ha dispuesto además un botón de *Read Config* con el que es posible enviar el mensaje *pedir configuración* de forma manual en cualquier instante, así como una opción para seleccionar las celdas que quieren forzar a descargarse (mediante *Force Balancing*).

(a) Pestaña *connect*.(b) Pestaña *main*.(c) Pestaña *config*.Figura 8.25: Capturas del *software* BMS Programmer.

### 8.3. Herramienta de procesamiento de datos BMS

Para poder analizar correctamente los datos del BMS capturados en un `.csv` con la herramienta *BMS Programmer GUI*, se ha desarrollado una herramienta de consola en Python denominada *BMS data processing* que permite, a partir de un `csv`, generar las siguientes gráficas:

- Gráfica de tensión de las celdas.
- Gráfica de tensión total de la batería.
- Gráfica de diferencia máxima de tensión entre celdas.
- Gráfica de todas las temperaturas medidas.
- Gráfica de la corriente medida.
- Gráfica del SOC.

Para ejecutar este *script* es necesario generar una determinada estructura de ficheros. En primer lugar, el *software* se localiza en la carpeta `src`, los datos se localizan en la carpeta `data` y las gráficas generadas son almacenadas en la carpeta `figures`. La llamada por consola se realiza empleando 3 argumentos: el nombre del fichero, el número de celdas en serie y el número de termistores. Así, por ejemplo, para procesar los datos del fichero llamado `test12092020.csv` de un BMS con 5 celdas y 9 termistores, se realiza la llamada como se muestra a continuación:

```
python3 bms-data-processing-d test12092020.csv 5 9
```

Esto generará una serie de gráficos, como se ha indicado anteriormente, en la carpeta `figure` que se nombran por su fecha (en formato *AAAADDMM*) seguida del tipo de gráfico (*batteryvoltage*, *cellvolt*, *soc* o *temp*).

#### 8.3.1. Código implementado

El código se ha dividido en tres ficheros diferentes: un fichero principal al que se ha denominado `bms-data-processing-d` y dos ficheros de librerías denominados `bms_data_plot_tools` y `bms_data_tools`.

Para la representación de las gráficas antes indicadas, es necesario, en primer lugar, la lectura de todos los parámetros del `csv`. Cada variable es almacenada en un array de `numpy`. Esto se realiza en el fichero principal y se muestra a continuación:

```

1 current = np.array ([])
2 voltage = np.array ([])
3 v_cells = np.array ([])
4 temp_cells = np.array ([])
5 SOC = np.array ([])
6 hour = np.array ([])
7 minute = np.array ([])
8 seconds = np.array ([])
9
10 print("Tratando de leer el fichero " + FILE_NAME)
11 with open(FILE_PATH+'data/'+FILE_NAME, newline='') as File:
12     reader = csv.reader (File)
13     for row in reader:
14         current = np.append(current , float(row[3]))
15         voltage = np.append(voltage , float(row[7+Ncell+Ntemp]))
16         SOC = np.append(SOC, float(row[5+Ncell+Ntemp]))
17         v_cells = np.append(v_cells , (row[4:4+Ncell]))
18         temp_cells = np.append(temp_cells ,(row[5+Ncell:5+Ncell+Ntemp]
19         ))
20         hour = np.append(hour , int(row[0][11:13]))
21         minute = np.append(minute , int(row[0][14:16]))
22         seconds = np.append(seconds , int(row[0][17:19]))
23         year = (row[0][6:10])
24         month = (row[0][3:5])
25         day = (row[0][0:2])
26         temp_cells = temp_cells.astype(float)
27         v_cells = v_cells.astype(float)

```

Listing 8.11: Inicialización y lectura de los valores desde el .csv con los datos del BMS.

Tras esto, se procede a la llamada de cada función de gráfico a la que se les pasa el nombre y la ruta del fichero donde se almacenará:

```

1 plot_name = year+month+day+"_temp"
2 plt.plot_bms_temp(FILE_PATH + '/figures/'+plot_name , temp_cells ,
3     day, month, year, hour, minute, seconds, Ntemp)
4
5 plot_name = year+month+day+"_temp"
6 plt.plot_bms_all_temp(FILE_PATH + '/figures/'+plot_name ,
7     temp_cells , day, month, year, hour, minute, seconds, Ntemp)
8
9 plot_name = year+month+day+"_cellvolt.png"
10 plt.plot_bms_cell_volt(FILE_PATH + '/figures/'+plot_name , v_cells
11     , day, month, year, hour, minute, seconds, Ncell)
12
13 plot_name = year+month+day+"_differencevolt.png"
14 plt.plot_bms_dif_cell_volt(FILE_PATH + '/figures/'+plot_name ,
15     v_cells , day, month, year, hour, minute, seconds, Ncell)

```

```

13 plot_name = year+month+day+"_differencevoltpercent.png"
14 plt.plot_bms_dif_cell_volt_percent(FILE_PATH + '/figures/' +
    plot_name, v_cells, day, month, year, hour, minute, seconds,
    Ncell)
15
16 plot_name = year+month+day+"_current.png"
17 plt.plot_bms_current(FILE_PATH + '/figures/' + plot_name, current,
    day, month, year, hour, minute, seconds)
18
19 plot_name = year+month+day+"_soc.png"
20 plt.plot_bms_soc(FILE_PATH + '/figures/' + plot_name, SOC, day,
    month, year, hour, minute, seconds)
21
22 plot_name = year+month+day+"_batteryvoltage.png"
23 plt.plot_bms_voltage(FILE_PATH + '/figures/' + plot_name,
    voltage, day, month, year, hour, minute, seconds)

```

Listing 8.12: Llamada a las funciones de representación de los datos del BMS.

Dichas funciones de generación de gráficas se localizan en la librería `bms_data_plot_tools`. A modo de ejemplo, se muestra el código de representación de la tensión de las celdas. El resto de funciones se han desarrollado de la misma forma y pueden consultarse en mayor detalle en el repositorio [46].

```

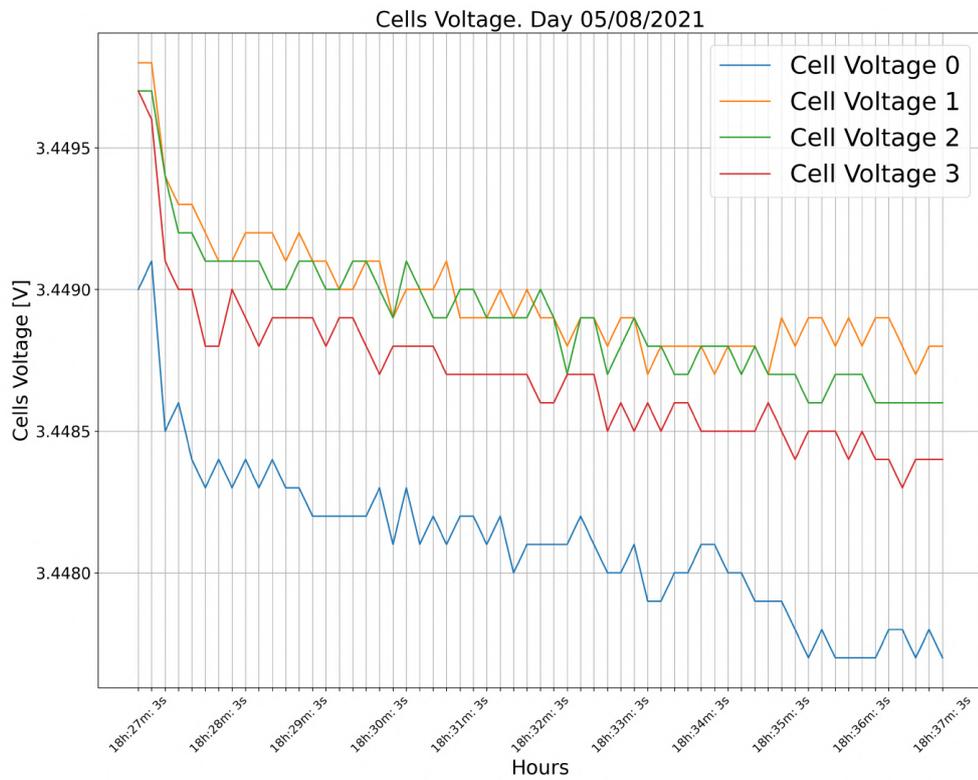
1 def plot_bms_cell_volt(plot_file_path, cellvolt, day, month, year
    , hour, minutes, seconds, Ncell):
2
3     tools.write_log(SCRIP_T_NAME, "Comenzando la preparacion del
    grafico %s." % plot_file_path)
4
5     W = 16
6     H = 12
7
8     fig, ax = plt.subplots(figsize=(W,H))
9
10    time = []
11
12    # Generamos la lista con las horas disponibles
13    for i in range(0, int(len(cellvolt)/Ncell)):
14        time.append("%2dh:%2dm:%2ds" % (hour[i], minutes[i], seconds[i]
    ))
15
16    for cell_index in range(Ncell):
17        aux = np.array([], dtype = float)
18        for i in range(0, int(len(cellvolt)/Ncell)):
19            aux = np.append(aux, cellvolt[i*Ncell + cell_index])
20        plt.plot(time, aux, '-', label = "Cell Voltage %d" % cell_index)
21        plt.xticks(rotation = 45, fontsize = 7)
22
23

```

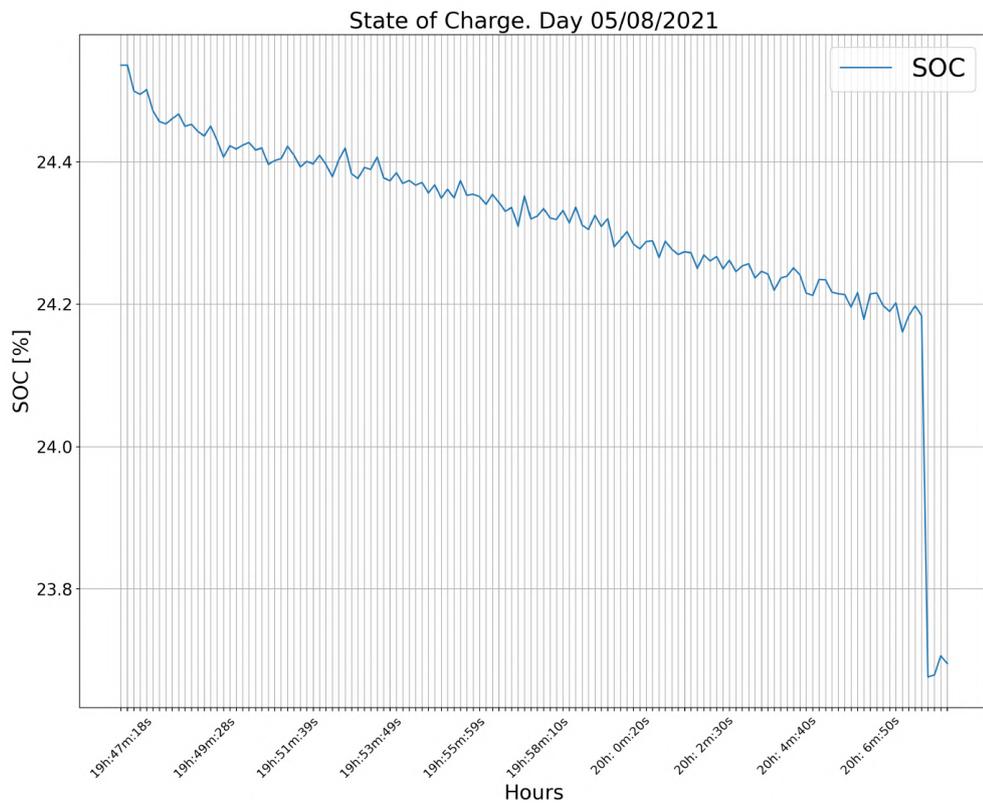
```
24 every_nth = round(len(seconds)/50)
25 for n, label in enumerate(ax.xaxis.get_ticklabels()):
26     if n % every_nth != 0:
27         label.set_visible(False)
28
29 plt.grid()
30 plt.xlabel('Hours')
31 plt.ylabel('Cells Voltage [V]')
32 plt.title('Cells Voltage. Day ' + day + '/' + month + '/' + year)
33 plt.legend()
34 plt.savefig(plot_file_path , dpi=300)
35 plt.show()
36 plt.close()
37 tools.write_log(SCRIPt_NAME, "El grafico %s ha sido procesado."%
    plot_file_path)
```

Listing 8.13: Función de representación de la tensión de las celdas.

Algunos ejemplos de las gráficas generadas se muestran en la Figura 8.26.



(a) Tensión de las celdas.



(b) SOC de la batería.

Figura 8.26: Ejemplos de gráficas generadas con los datos del BMS.

## 8.4. Software SOC Estimation

Para el análisis de los datos de los ensayos de las celdas, se ha diseñado un *script* en python para ser ejecutado desde consola, denominado `estimate_SOC`. Presenta una estructura de archivos similar al anterior (código localizado en `src`, datos `csv` en la carpeta `data` y las gráficas generadas en la carpeta `figures`). Para la ejecución de este *script* es necesario pasarle como argumentos el nombre del fichero `csv`, el grado de la aproximación polinómica y la resistencia interna (en  $\Omega$ ). Así, para el fichero con nombre `DL_recordata.csv`, ejecutando una aproximación de grado 4 y con una resistencia interna de  $0.0066 \Omega$ , debe lanzarse con el siguiente comando:

```
python3 estimate_SOC.py DL_Recordata.csv 4 0.0066
```

La estructura principal del código también es muy similar al anterior. En primer lugar, se realiza la lectura de los parámetros desde el `csv` para almacenarlos en arrays de `numpy`. Posteriormente, se procede a calcular el  $V_{oc}$  y  $SOC$  para graficar estos dos valores haciendo uso de la función `plot_SOC_data()`. Tras esto, se calcula la regresión polinómica desde grado 2 hasta grado 9 haciendo uso del comando `polyfit` de `matplotlib`, generando para cada uno de ellos una gráfica y una salida por consola de los coeficientes de la curva (como muestra la Figura 8.27). Estas figuras son almacenadas en la carpeta `figures` con el nombre `SOC_polynomial` seguido del grado calculado en la aproximación. También se generan otras figuras similares en las que se cuantiza el error máximo cometido con la aproximación y que se denominan `SOC_regretion`. Estas gráficas ya fueron visualizadas en el Capítulo 6, concretamente, en las Figuras 6.7 y 6.8.



```
augusto@augusto-Modern-14: ~/Documentos/tfm_augusto/Software/SOC_estimation/src
Archivo Editar Ver Buscar Terminal Ayuda
Coeficientes calculados para grado 4 :
[ -1.04275746  12.06683149 -51.53767802  97.08832862 -68.45442121]
La expresion polinomial es:
-1.0428*v^(4) + 12.0668*v^(3) + -51.5377*v^(2) + 97.0883*v^(1) + -68.4544
```

Figura 8.27: Salida por consola de la aproximación polinómica.

## 8.5. Software de la Herramienta de cálculo de consumos del nodo

Para la determinación de los consumos se ha desarrollado, por un lado, un servicio del sistema y por otro una herramienta de ordenación y representación

de datos. El servicio del sistema se ha empaquetado dentro de un paquete `.deb`<sup>1</sup>, para facilitar su instalación en sistemas Linux basados en Debian.

A continuación, se muestra de forma detallada cada una de las herramientas.

### 8.5.1. Capturadora de consumos en el nodo

El dispositivo elegido como equipo de adquisición de sonido es una Raspberry Pi 3B+ con el sistema adquirente que se detalló en el Capítulo 3. A este sistema se le ha añadido un sensor de corriente, denominado **INA 219**. Este sensor precisa de una pequeña resistencia en la que se mide la caída de tensión, y a partir de ella determina la corriente que circula. Además, permite medir el voltaje total del sistema y posee comunicación *I2C*. Este circuito se ha conectado a la Raspberry mediante sus pines de *I2C* (localizados en los pines 3 y 5).

El servicio del sistema se ha desarrollado en Python y se ha programado su ejecución empleando *systemd*<sup>2</sup>. Cada 5 minutos, es lanzado el servicio, de forma que se lee la corriente consumida, la tensión, la potencia instantánea y el uso del *core*. Estos datos se almacenan en un fichero *.json* para cada día. En la Figura 8.28, se muestra una captura de un fragmento del fichero *json* generado.

```

▼ data:
  ▼ 0:
    year:          "2021"
    month:         "04"
    day:           "12"
    hour:          "12"
    minutes:       "52"
    seconds:       "1618228332"
    current [mA]:  -508.3536585365853
    voltage [V]:   12.18
    power [mW]:    5689.0243902439015
    Temp. CPU [C]: 60.148

```

Figura 8.28: Fragmento del fichero *json* generado.

Dado que esta herramienta se ejecuta de forma permanente en el nodo, es preciso idear un sistema de ordenación de los datos. Para ello, se ha creado una estructura de ficheros que facilite su interpretación.

<sup>1</sup>*.deb* es la extensión del formato de los paquetes de *software* de la distribución Linux Debian y derivados.

<sup>2</sup>*systemd* es un conjunto de *daemons* de administración del sistema, bibliotecas y herramientas diseñados como una plataforma de administración y configuración central para interactuar con el núcleo del sistema operativo GNU/Linux.

En el directorio `usr/share`, la primera vez que se lanza el servicio se crea una carpeta denominada `power-data`. En su interior se localizan 3 directorios diferentes: `data`, `src` y `log`. En el primero de ellos se almacenan los datos, para ello, se crea una subcarpeta con el año, otra con el mes, y finalmente otra con el día; dentro de esta última es donde se almacena el fichero `json` de todo el día, con una entrada cada 5 minutos.

En el directorio `log` se localiza un fichero llamado `power-data.log` en el que se muestran los diferentes `logs` de ejecución del servicio. De esta forma es posible conocer si ha habido un error durante el funcionamiento normal de la herramienta. Finalmente, el directorio `src` contiene los códigos de python que se ejecutan. Esta estructura de ficheros implementada se muestra en la Figura 8.29.

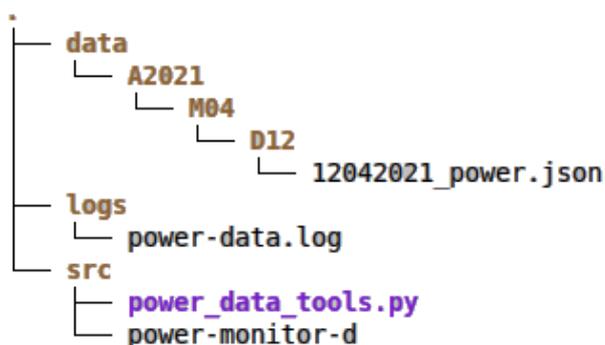


Figura 8.29: Estructura de ficheros generada.

En el directorio `/lib/systemd/system` se localiza el fichero de configuración del lanzamiento del servicio (`power-monitor-d.service`) y el `timer` (`power-monitor-d.timer`). La configuración de estos ficheros es la que se muestra a continuación:

#### **power-monitor-d.service**

```

1 [Unit]
2 Description= Power data capturer for smart nodes (power_monitor
   service)
3
4 [Service]
5 Type=oneshot
6 ExecStart=/usr/bin/python3 /usr/share/power-data/src/power-
   monitor-d
7 #ExecStop=/bin/kill -SIGINT ${MAINPID}
8 Environment=PYTHONUNBUFFERED=1
9

```

```

10 [ Install ]
11 WantedBy=default.target

```

Listing 8.14: Configuración del servicio *power-monitor*.

### **power-monitor-d.timer**

```

1 [ Unit ]
2 Description= Timer that executes power-monitor-d.service every 5
  minutes
3 Requires=power-monitor-d.service
4
5 [ Timer ]
6 OnUnitActiveSec=5min
7 OnBootSec=1min
8 Unit=power-monitor-d.service
9
10 [ Install ]
11 WantedBy=multi-user.target

```

Listing 8.15: Configuración del *timer* del servicio *power-monitor*.

El código desarrollado se ha dividido, como en casos anteriores en un fichero python principal (*power-monitor-d*) y un fichero de librerías (*power\_monitor\_tools*). Este código puede verse en detalle en el GitHub del proyecto [46]. Finalmente, se ha generado el paquete *.deb* haciendo uso de la herramienta *debuild*.

## **8.5.2. Herramienta de representación de datos**

Para la representación de los datos de consumo captados por la herramienta *power-monitor*, se ha desarrollado una utilidad llamada *power-data-processing*. Esta, a partir de la estructura de ficheros de *año*, *mes*, *día* antes descrita, genera los gráficos correspondientes a cada día y los almacena en el directorio correspondiente.

Para conocer el estado de ejecución de dicha herramienta, dispone de dos ficheros de *logs*, en los que se muestra, por un lado el estado de ejecución del programa (en el *power-data.log*) y por otro, el nombre de aquellos gráficos que ya han sido procesados (en el *processed\_plot.log*). De esta forma, al lanzar nuevamente la utilidad, solo se representan aquellos gráficos correspondientes a los días que no estén en el fichero *processed\_plot*, evitando la ejecución redundante. Finalmente, la estructura de ficheros resultante se muestra en la Figura 8.30.

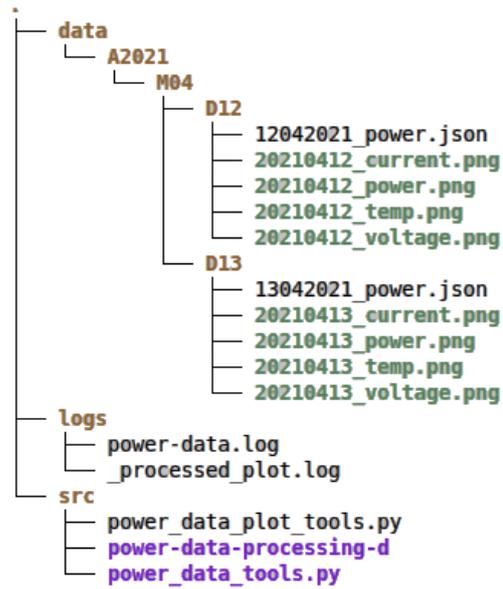


Figura 8.30: Estructura de ficheros generada.

Las gráficas generadas y almacenadas se han mostrado en el Capítulo 6, concretamente, en las Figuras 6.2 y 6.4. No obstante, también pueden consultarse en el Anexo C.



# Capítulo 9

## Pruebas del Sistema

En el presente capítulo se muestran las diferentes pruebas de carga y descarga que se han realizado al conjunto de la batería y el BMS. Se mostrarán las gráficas generadas con la herramienta `bms-data-processing` a partir de los datos capturados mediante el *software BMS Programmer*, con el módulo CAN2Serial conectada al sistema de batería y al *equipo portátil de diagnóstico*. Todas estas gráficas pueden verse con un mayor detalle en el Anexo D.

### 9.1. Metodología de las pruebas realizadas

Para realizar los diferentes *tests* de descarga de la batería, se ha empleado la carga electrónica Rigol descrita en el Capítulo 2. Se ha ajustado la carga a valores constantes de resistencia de 44  $\Omega$ , 11 $\Omega$  y posteriormente, a una corriente constante de 10A. Empleando la herramienta desarrollada *BMS CAN Programmer* conectada al bus CAN a través del módulo CAN2Serial, se ha procedido a monitorizar y grabar en tiempo real los datos en un fichero `.csv`. Posteriormente, estos datos se han procesado con la utilidad *BMS Data Processing* para generar los gráficos que se comentan en este capítulo. En la Figura 9.1 puede verse un diagrama del montaje expuesto.

De igual forma, las pruebas de carga se han realizado con el mismo montaje a excepción de la carga Rigol que ha sido sustituida por una fuente de corriente de laboratorio capaz de proporcionar 15A de forma constante en el rango de tensiones de la batería. Nuevamente, los datos se han almacenado en un `.csv`, se han procesado y se presentan a lo largo de las siguientes secciones.

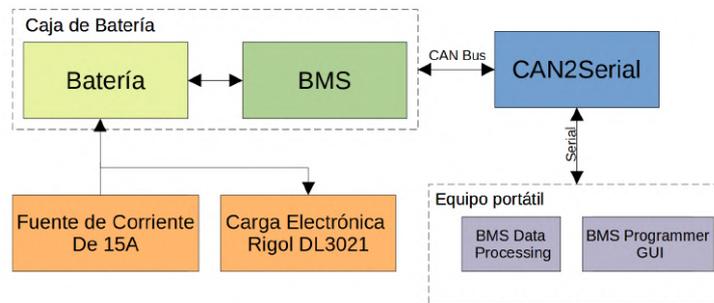


Figura 9.1: Esquema del montaje adoptado para las pruebas de carga y descarga.

## 9.2. Prueba de descarga a $44 \Omega$

La primera prueba realizada ha sido una descarga con la carga electrónica comportándose como una resistencia de  $44 \Omega$ . Dado que la tensión en esta prueba era de 13.8 V, la corriente de descarga esperada es de 314 mA. Se procede a realizar esta descarga durante un período de 10 minutos.

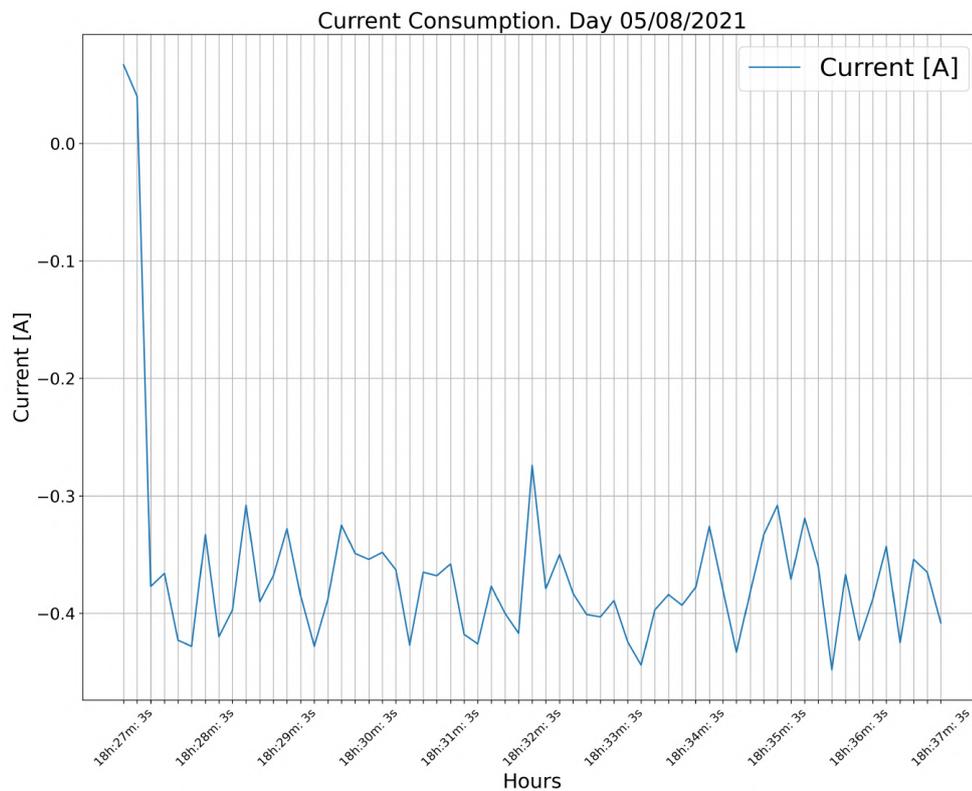


Figura 9.2: Corriente de descarga medida en el *test* de  $44\Omega$  (Anexo D.1).

En la Figura 9.2 se puede observar la corriente de descarga monitorizada. Puede observarse el ruido en la medición de la misma. Este ruido se debe principalmente a estar midiendo una corriente pequeña en comparación con el fondo de escala de 30 A. Se comprueba que el criterio de signos es correcto, dado que la corriente de descarga es medida como negativa.

La Figura 9.3 muestra la tensión de cada celda de la batería. Puede observarse una caída de tensión de aproximadamente 0.5 mV debido a la resistencia interna de las celdas en el momento que se conecta el consumo. En esta prueba, al solicitarse muy poca corriente, la tensión de las celdas cae menos de 1 mV. No obstante, puede observarse como este decremento de tensión presenta la misma tendencia en todas las celdas, garantizándose así que las celdas se mantienen balanceadas. Esto puede verse con mejor detalle en la gráfica de máxima diferencia de tensión entre celdas (Figura 9.4).

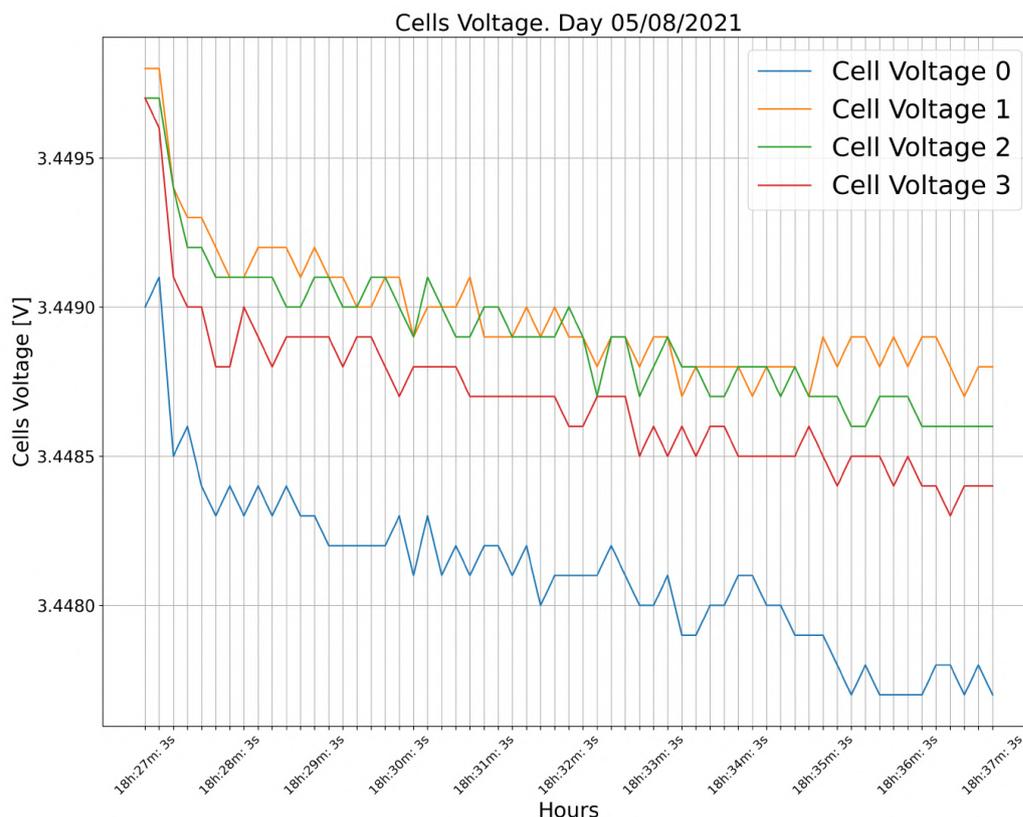


Figura 9.3: Voltajes de las celdas medidos en el *test* de 44 $\Omega$  (Anexo D.2).

La gráfica de la Figura 9.4 muestra la diferencia máxima entre celdas en cada instante. Puede verse como se parte de una diferencia inferior a 1 mV y esta diferencia se mantiene durante toda la prueba de descarga, reduciéndose a medida que el BMS va balanceando hasta alcanzar prácticamente una diferencia inferior a  $100 \mu V$ .

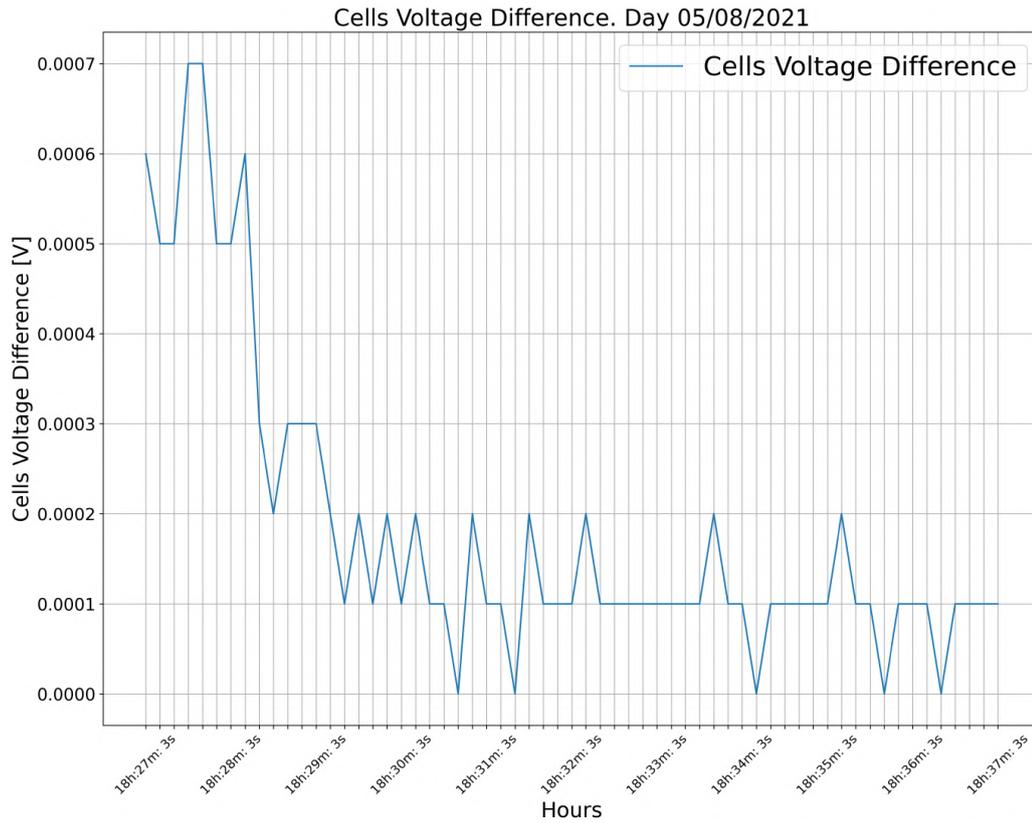


Figura 9.4: Diferencia máxima entre celdas durante el *test* de  $44\Omega$  (Anexo D.3).

El Estado de carga de la batería corresponde a un valor de aproximadamente 24.3% y se reduce ligeramente a medida que se produce la descarga de la batería hasta alcanzar un valor de 24.225%, como se muestra en la Figura 9.5.

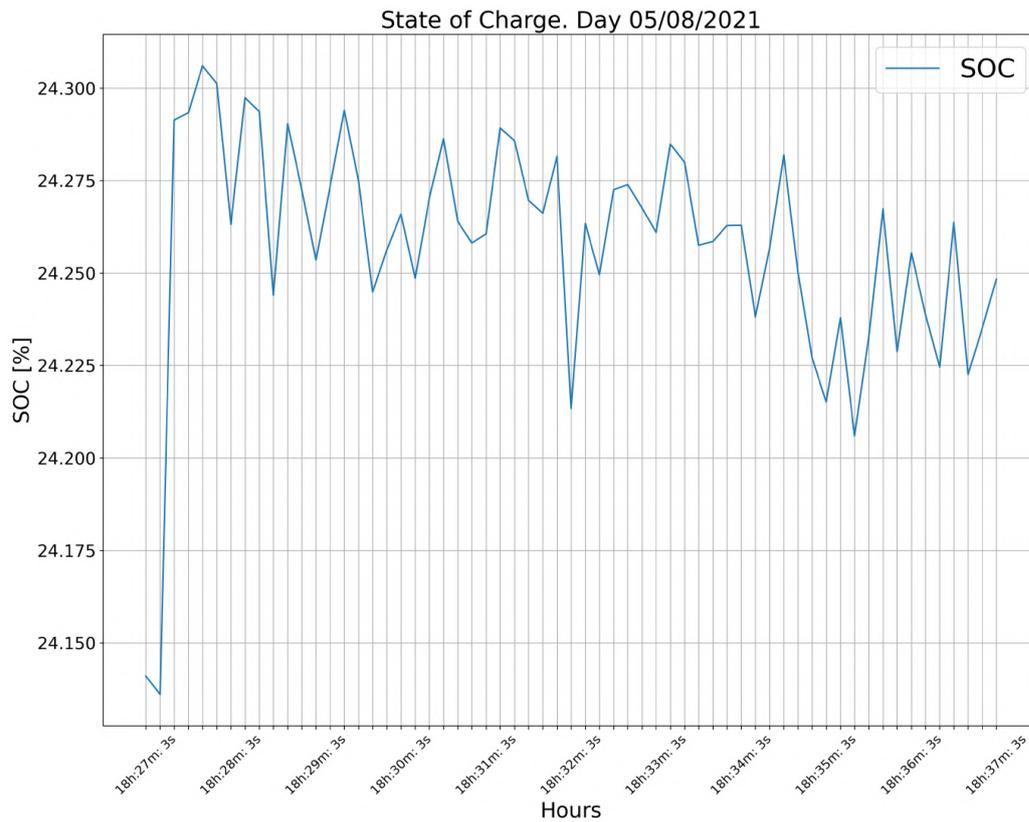


Figura 9.5: Estado de Carga de la batería durante el *test* de 44 $\Omega$  (Anexo D.4).

Finalmente, se muestra la temperatura medida por los cuatro termistores instalados en el acumulador en la Figura 9.6. Puede observarse como las mediciones se mantienen constantes a lo largo del tiempo, siendo el valor promedio de temperatura de 26.75°C.

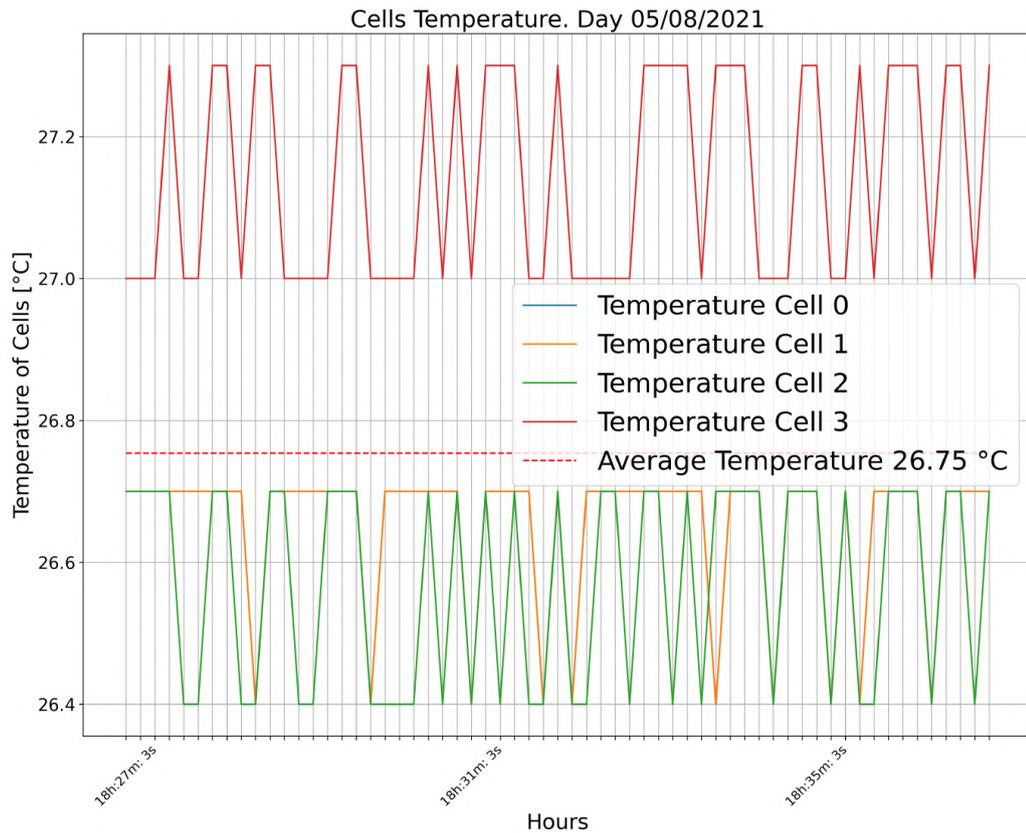


Figura 9.6: Temperaturas medidas durante el *test* de 44Ω (Anexo D.5).

### 9.3. Prueba de descarga a 11 $\Omega$

La segunda prueba realizada ha sido una descarga con 11  $\Omega$ . En este caso la corriente de descarga esperada es de 1.25A. Se procede a realizar esta descarga durante un período de 20 minutos.

En la Figura 9.7 se puede observar la corriente de descarga monitorizada. Puede observarse como el ruido en la medición de la misma se ha reducido con respecto a la prueba anterior, al medir un valor de corriente mayor. En este caso la corriente oscila entre -1.19 A y -1.28 A (en descarga).

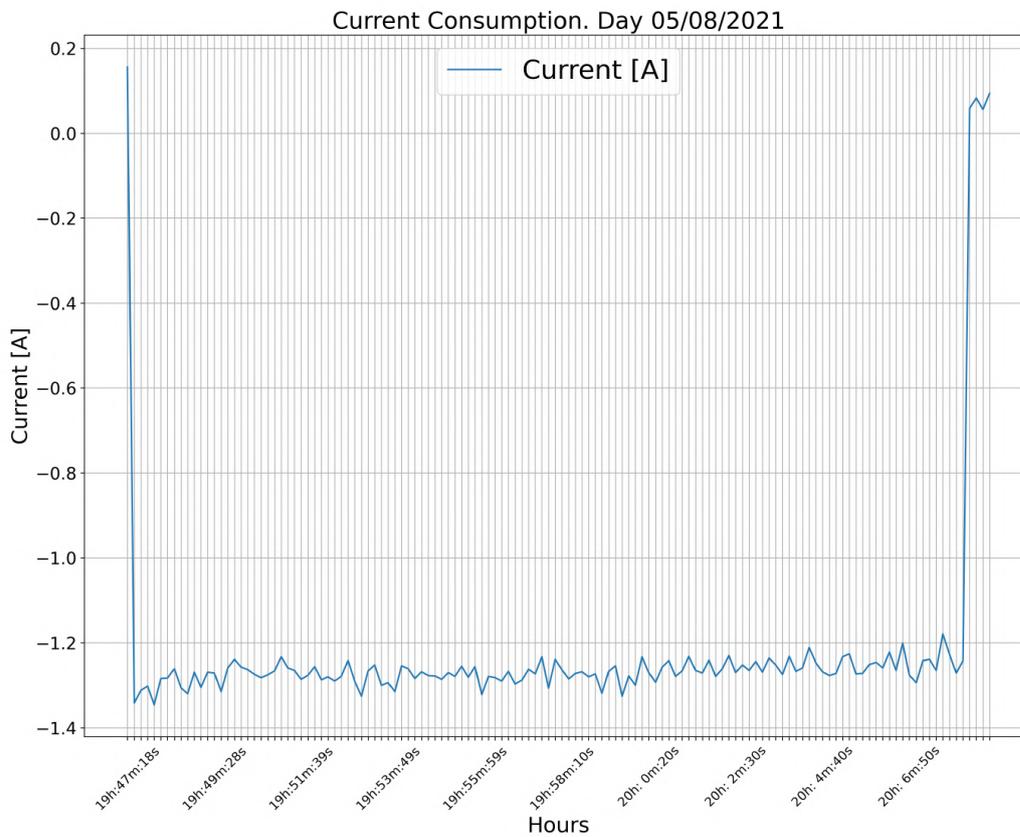


Figura 9.7: Corriente de descarga medida en el *test* de 11 $\Omega$  (Anexo D.6).

La Figura 9.8 muestra la tensión de cada celda de la batería. Puede observarse que la caída de tensión en este caso debido a la resistencia interna de las celdas es de 1.9 mV. Cabe destacar como el decremento de tensión de todas las celdas presenta prácticamente la misma pendiente, demostrándose el correcto funcionamiento del algoritmo de balanceo implementado. Las celdas, durante esta prueba se han descargado un total de 4 mV, manteniéndose la diferencia entre celdas inferior a 2 mV en todo momento.

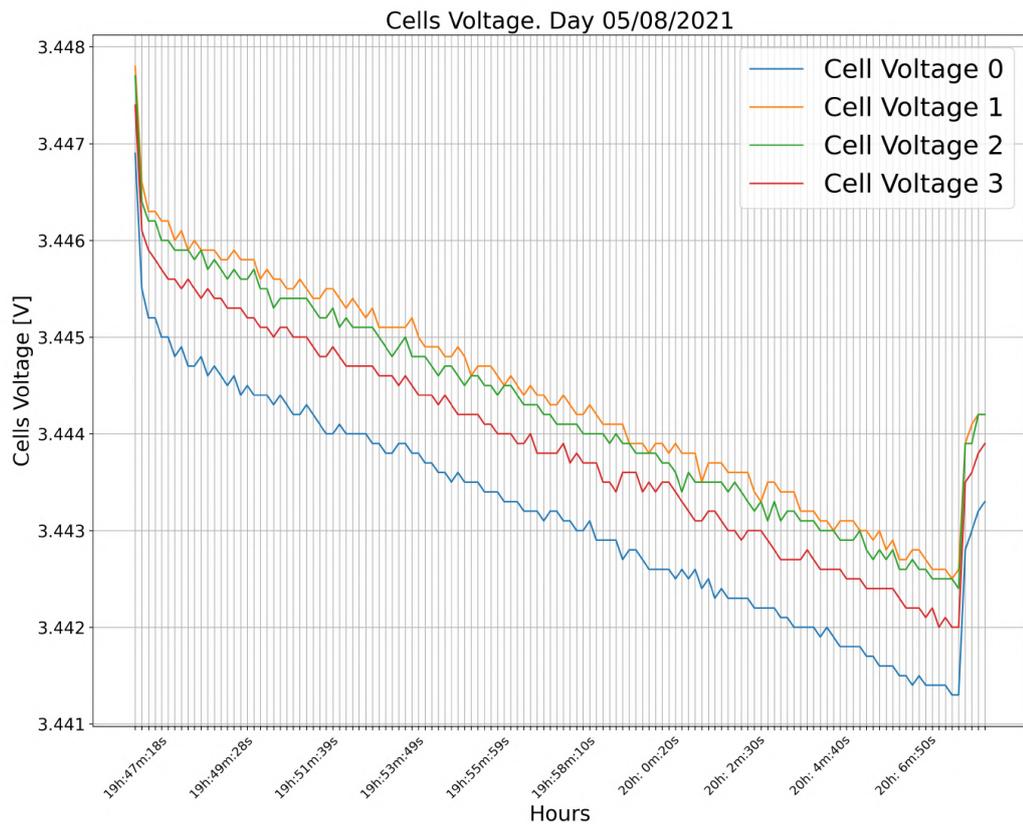


Figura 9.8: Voltajes de las celdas medidos en el *test* de 11Ω (Anexo D.7).

La gráfica de la Figura 9.9 muestra la diferencia máxima entre celdas en cada instante. En esta prueba, en todo momento esta diferencia se ha mantenido en un valor inferior a 2.5 mV.

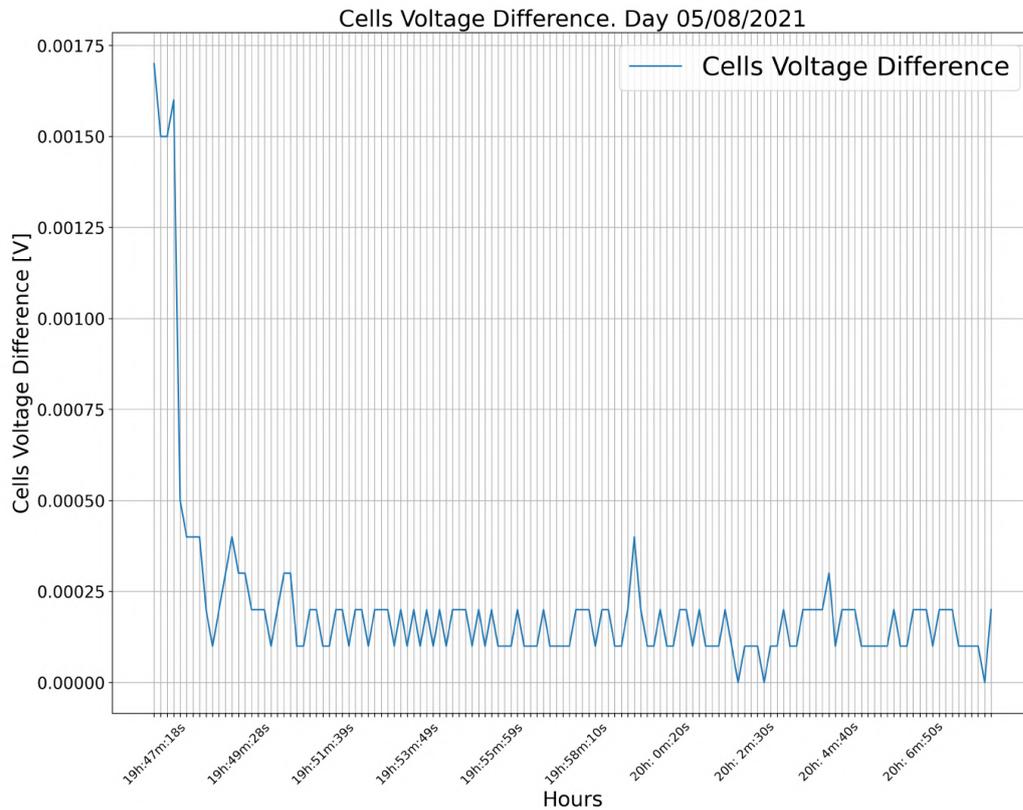


Figura 9.9: Diferencia máxima entre celdas durante el *test* de  $11\ \Omega$  (Anexo D.8).

El Estado de carga de la batería comienza en un valor de aproximadamente 24.5% y se reduce a medida que se produce la descarga de la batería hasta alcanzar un valor de 24.2%, como se muestra en la Figura 9.10.

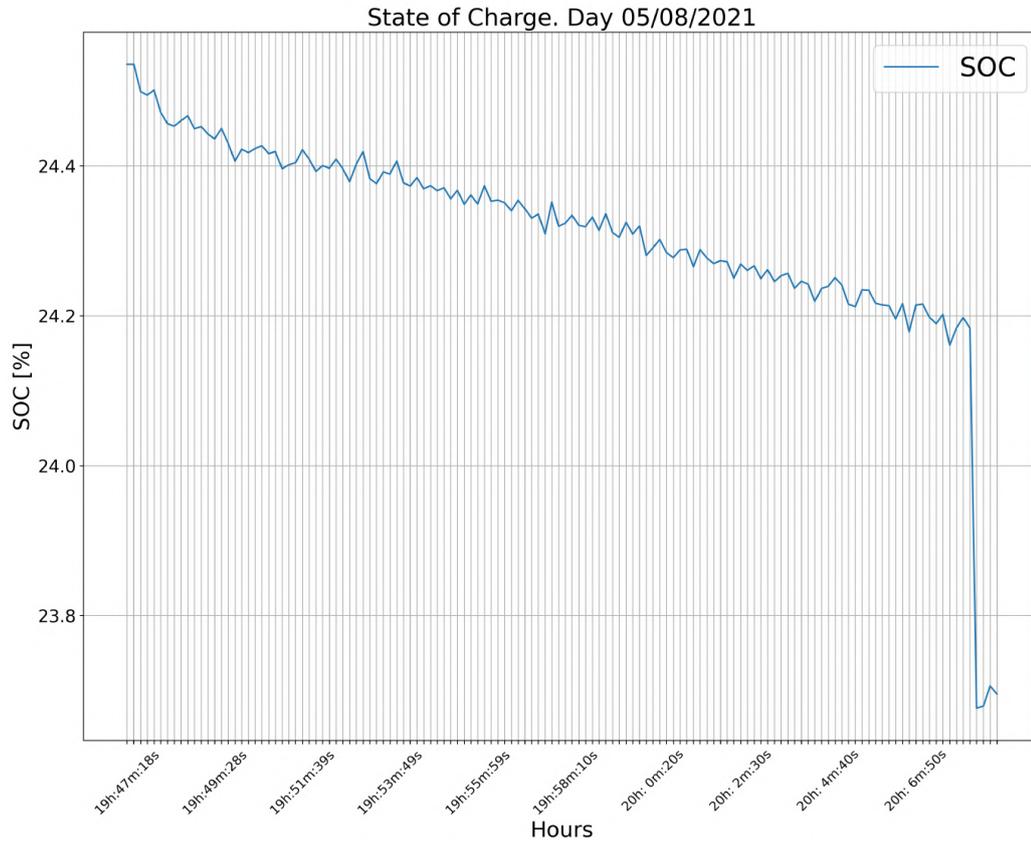


Figura 9.10: Estado de carga durante el *test* de  $11\Omega$  (Anexo D.9).

Finalmente, se representa en la Figura 9.11 la temperatura. A medida que evolucionó el ensayo, la temperatura en el *termistor 3* (localizado en el borne positivo) aumentó ligeramente (un valor de  $1.2^\circ\text{C}$ ).

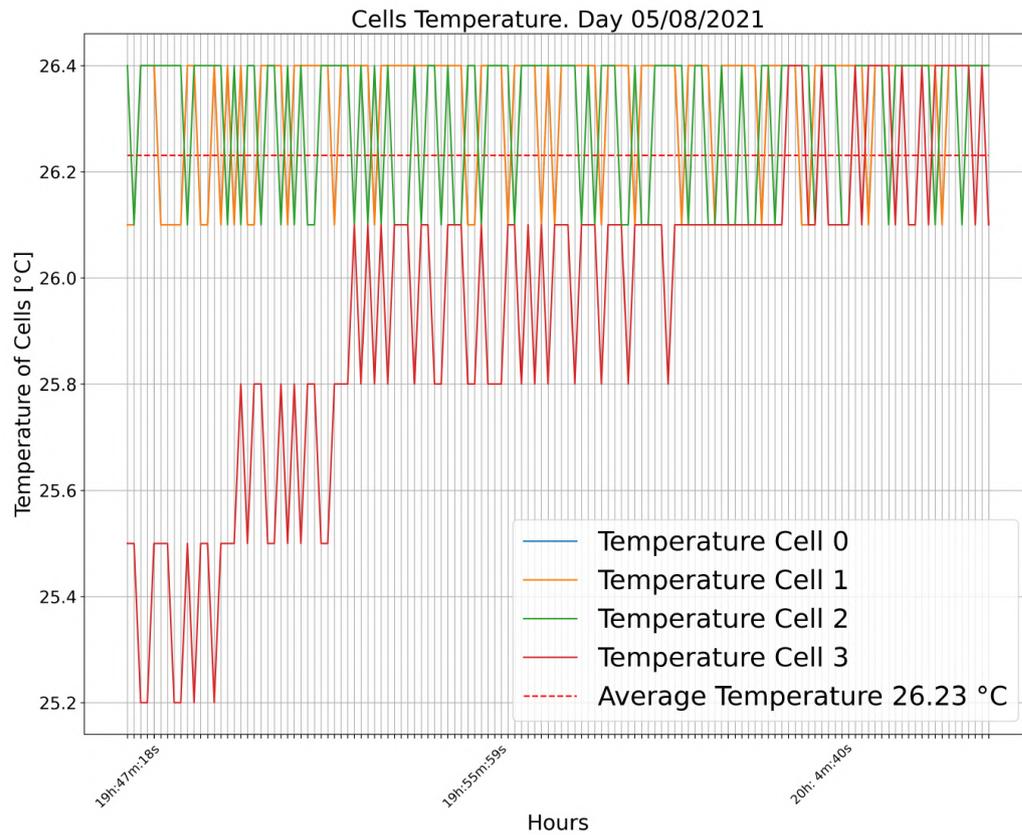


Figura 9.11: Temperaturas medidas durante el *test* de  $11\ \Omega$  (Anexo D.10).

## 9.4. Prueba de descarga a 10 A

La última prueba de descarga realizada ha sido a una corriente constante de 10 amperios. Se procede a realizar esta descarga durante un período de 90 minutos.

En la Figura 9.12 se puede observar la corriente de descarga monitorizada. Esta se mantuvo aproximadamente constante en los 10 A a medida que la tensión de la batería iba decayendo. Puede observarse, al inicio y final de la gráfica, la partida y retorno a reposo de la batería durante el ensayo.

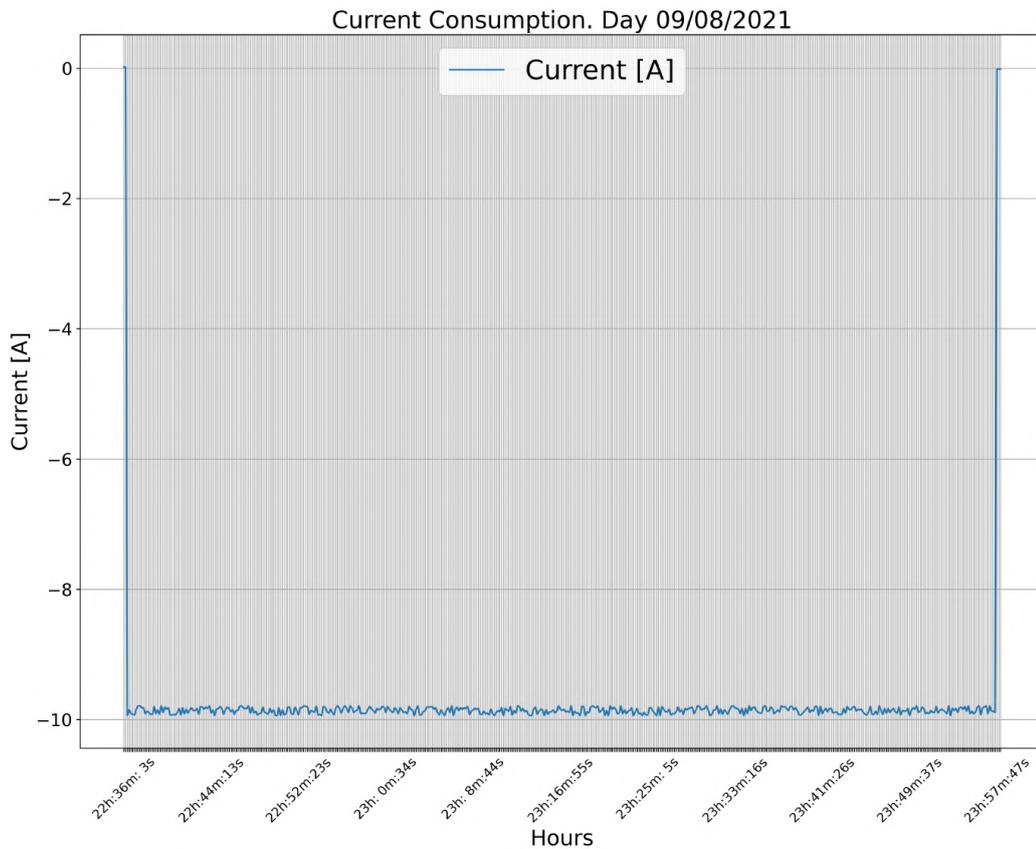


Figura 9.12: Corriente de descarga medida en el *test* a 10A (Anexo D.11).

La Figura 9.13 muestra la tensión de cada celda de la batería. Nuevamente, la reducción de tensión de todas las celdas presenta prácticamente la misma pendiente, demostrándose así el correcto funcionamiento del balanceo pasivo del BMS. Las celdas, durante esta prueba se han descargado un total de 60 mV.

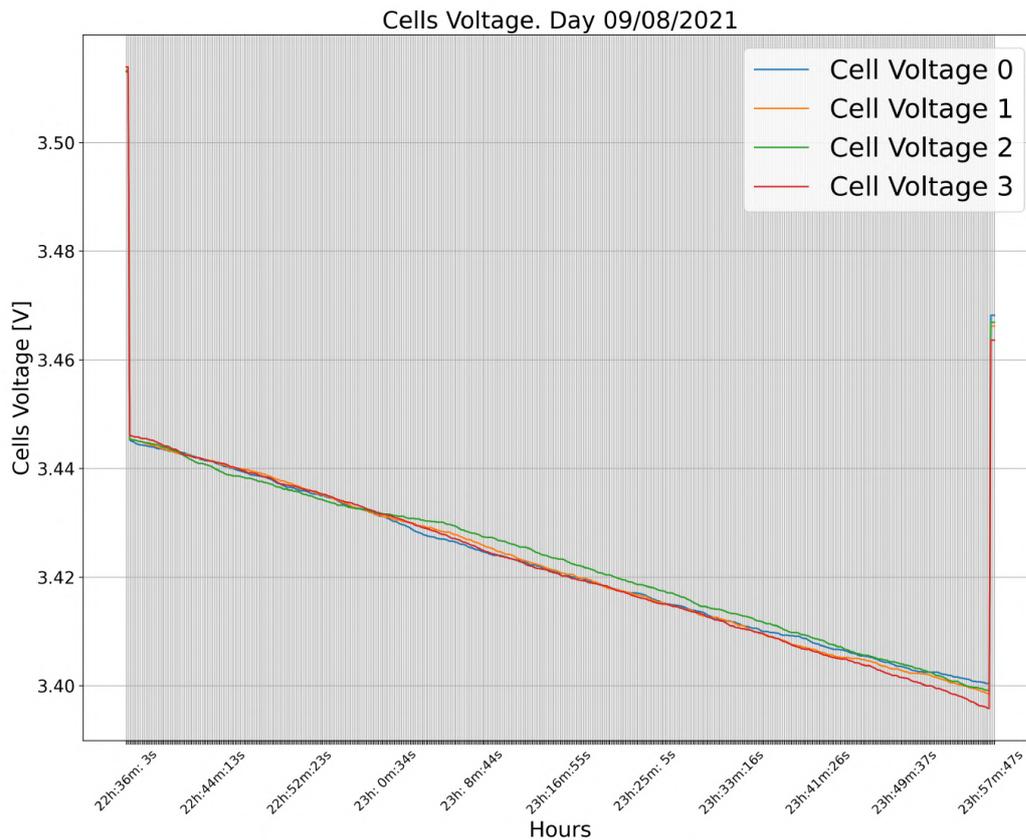


Figura 9.13: Voltajes de las celdas medidos en el *test* a 10A (Anexo D.12).

Tras esto, se representa la evolución de la tensión de la batería en la Figura 9.14. Cabe destacar que los *escalones* de la gráfica son debidos a la codificación de los mensajes CAN, que hacen que el incremento mínimo sea de 10 mV. La tensión evoluciona desde los 13.78 V en el instante inicial hasta los 13.59 V al finalizar el ensayo 91 minutos después.

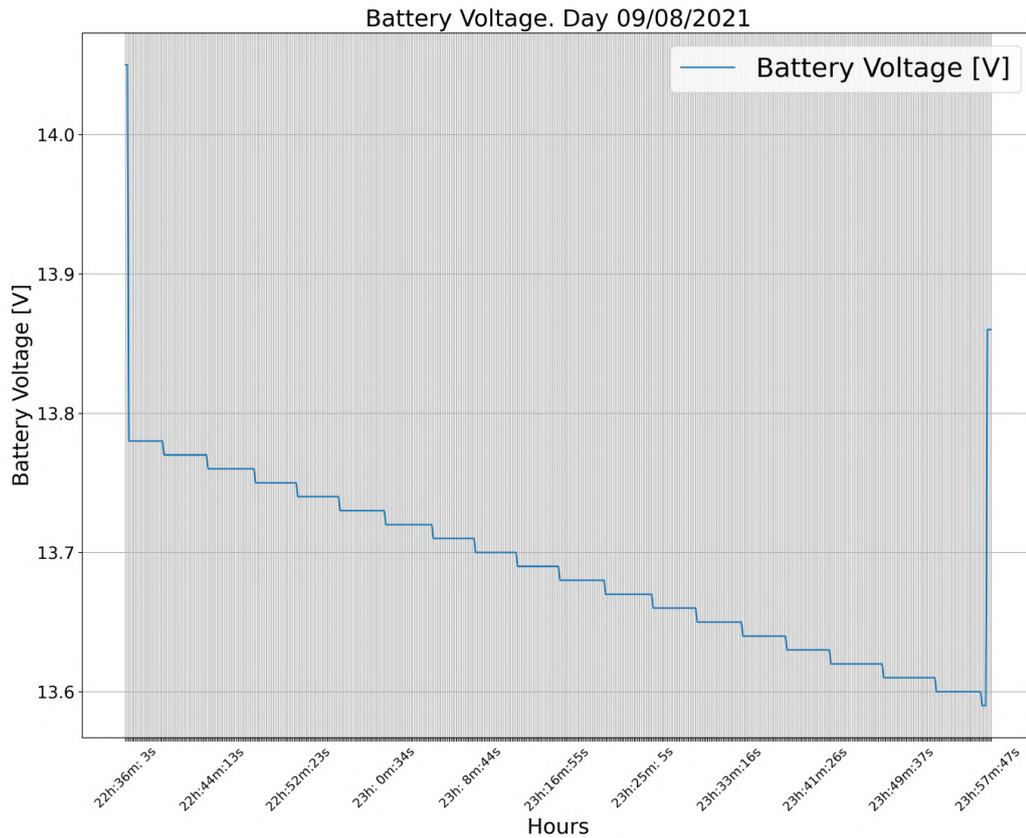


Figura 9.14: Voltajes de la batería en el *test* a 10A (Anexo D.15).

La gráfica de la Figura 9.15 muestra la diferencia máxima entre celdas en cada instante. Esta diferencia, ha sido en todo momento prácticamente 0. Esto es debido a que desde las pruebas anteriores han pasado aproximadamente 24h y el sistema ha estado balanceando las celdas hasta llegar a una diferencia inferior a 1 mV (consigna establecida en la configuración).

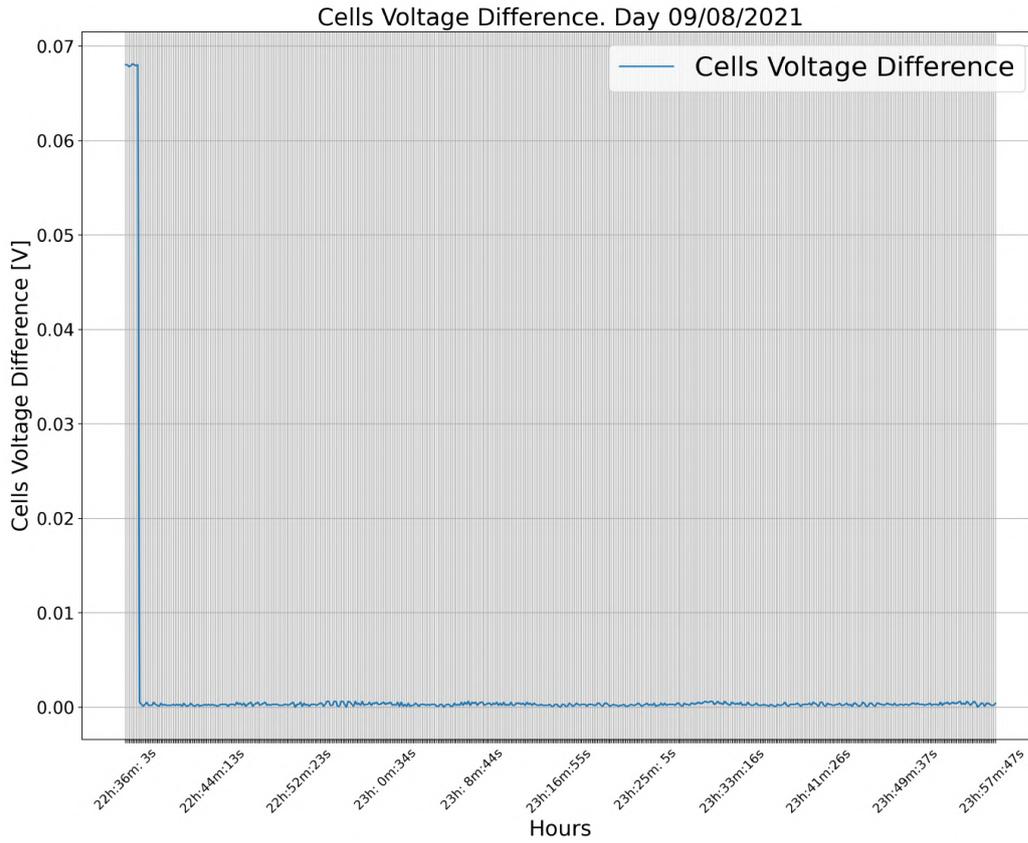


Figura 9.15: Diferencia máxima entre celdas durante el *test* de 10A (Anexo D.13).

El Estado de carga de la batería comienza en un valor de aproximadamente 29% y se reduce a medida que se produce la descarga de la batería hasta alcanzar un valor de 25.4%, como se muestra en la Figura 9.16.

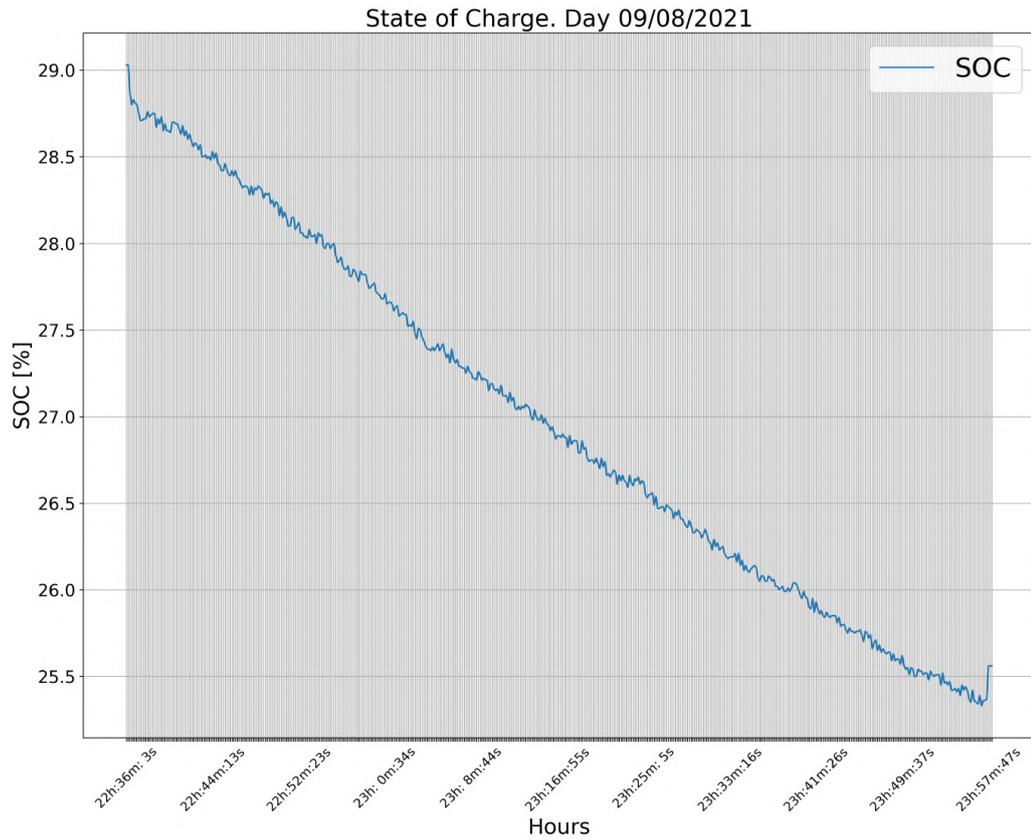


Figura 9.16: SOC durante el *test* de 10A (Anexo D.14).

Finalmente, se representa en la Figura 9.17 la temperatura. Durante esta prueba, la temperatura de la batería se ha mantenido aproximadamente constante en un valor de 26.5°C.

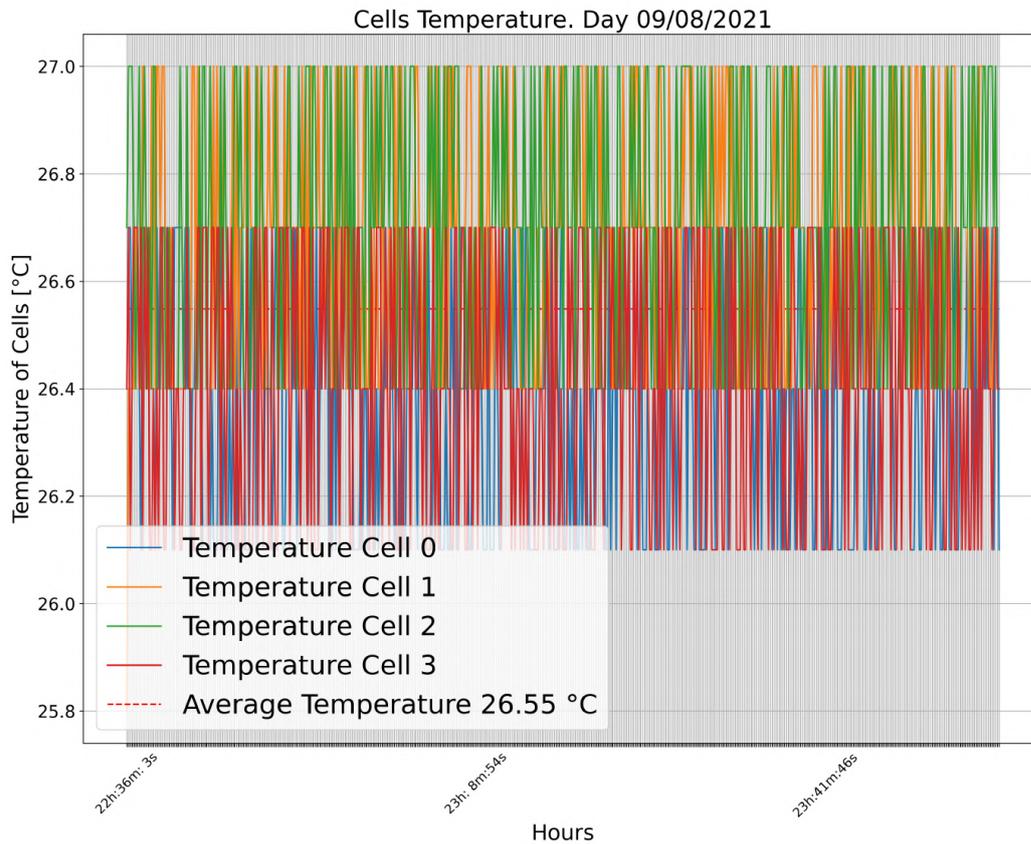


Figura 9.17: Temperaturas medidas durante el *test* de 10A (Anexo D.16).

## 9.5. Prueba de carga a 15A

Tras finalizar la prueba de descarga de 10 A, se procede a cargar la batería a una corriente constante de 15 A. Este proceso de carga ha durado aproximadamente 2 horas.

En la Figura 9.18 se puede observar la corriente de carga de la batería monitorizada. Esta medición se mantuvo aproximadamente constante a 15 A con un error máximo de lectura de 140 mA.

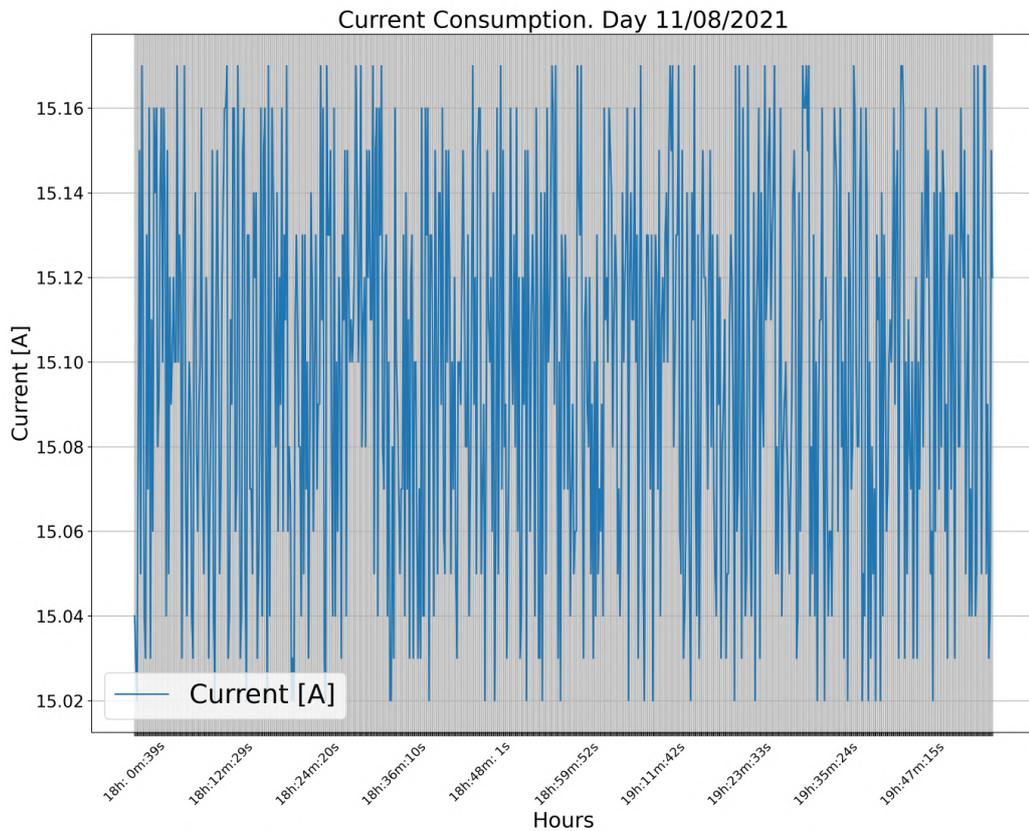


Figura 9.18: Corriente de carga medida en el *test* a 15A (Anexo D.17).

La Figura 9.19 muestra la tensión de cada celda de la batería. Se observa, al igual que durante la descarga, que la tensión de las diferentes celdas crece con la misma pendiente, por tanto, el balanceo está funcionando correctamente. Las celdas, durante esta prueba se han cargado un total de 70 mV.

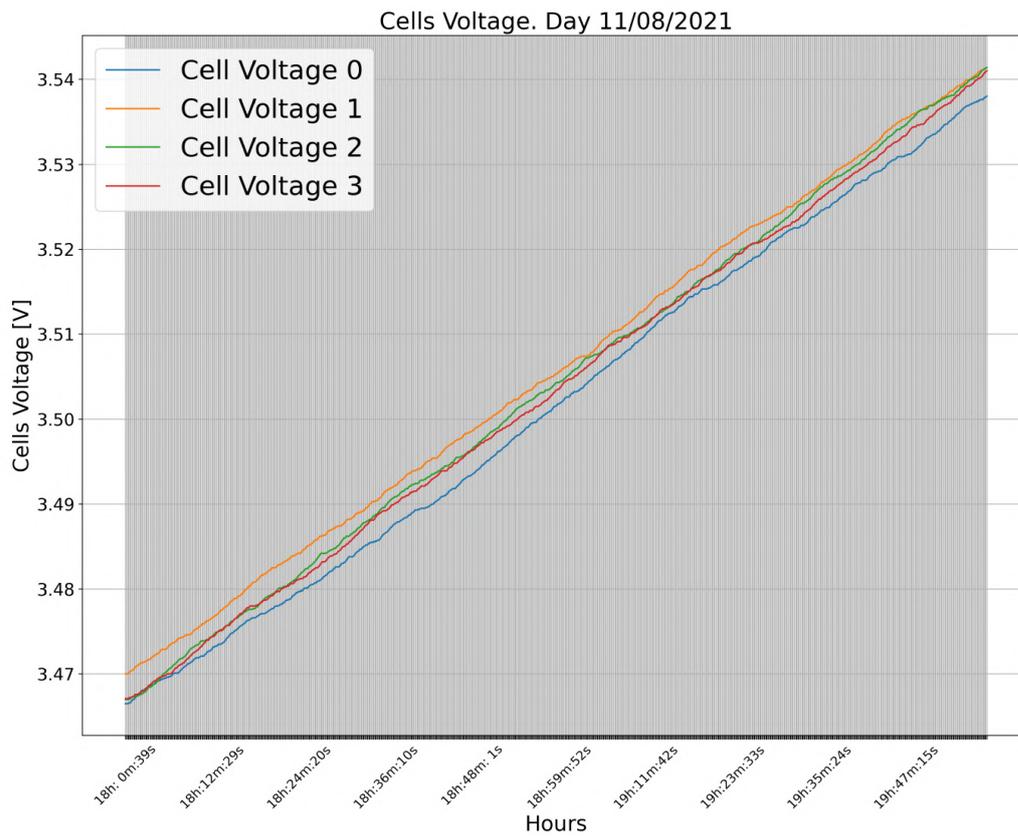


Figura 9.19: Voltajes de las celdas medidos en el *test* a 15A (Anexo D.18).

Tras esto, se representa la evolución de la tensión de la batería en la Figura 9.20. La tensión evoluciona desde los 13.84 V en el instante inicial hasta los 14.2 V al finalizar el ensayo de carga.

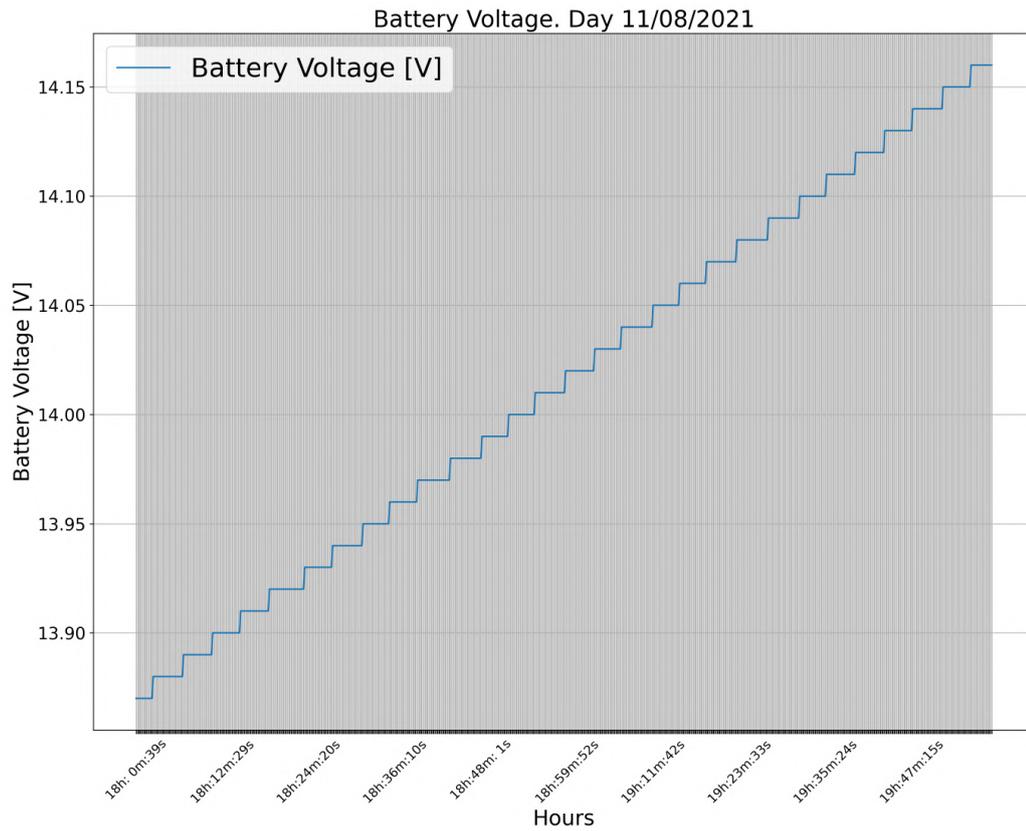


Figura 9.20: Voltajes de la batería en el *test* a 15A (Anexo D.21).

La gráfica de la Figura 9.21 muestra la diferencia máxima entre celdas en cada instante. Esta diferencia ha sido como máximo  $600 \mu V$  en todo momento, inferior a 1 mV establecido como diferencia máxima en la configuración.

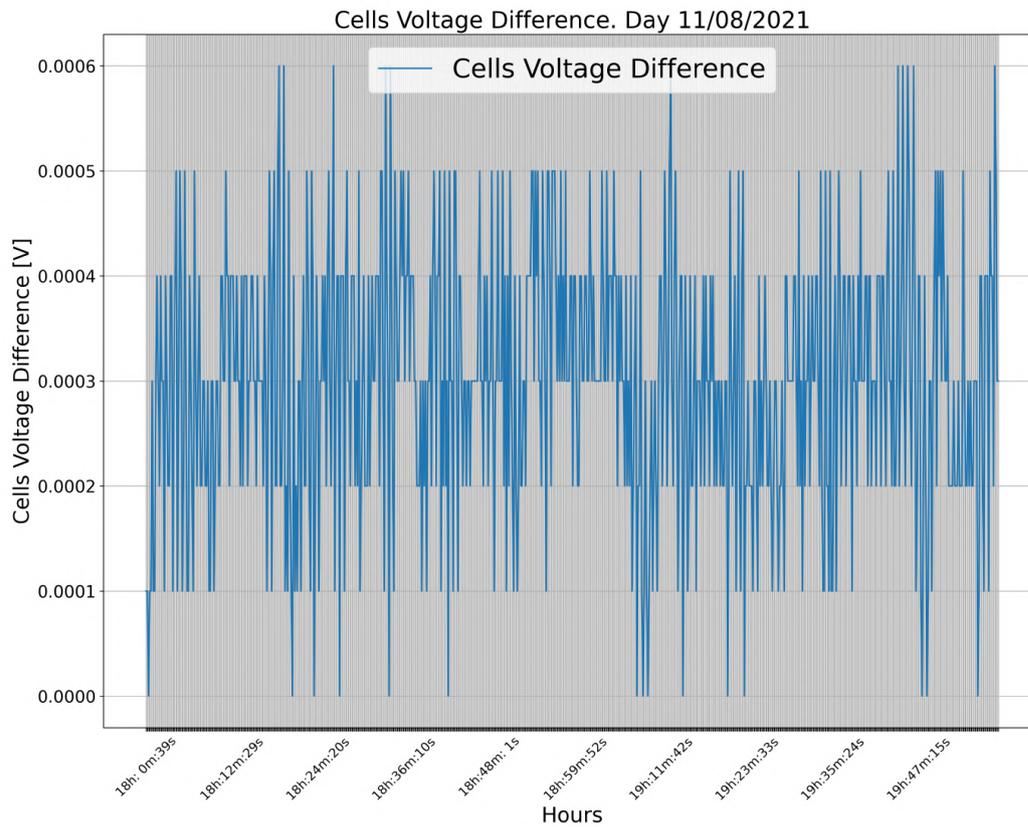


Figura 9.21: Diferencia máxima entre celdas durante el *test* de 15A (Anexo D.19).

El Estado de carga de la batería comienza en un valor de aproximadamente 18.5% y se incrementa a medida que se produce la carga de la batería hasta alcanzar un valor de 23.5%, como se muestra en la Figura 9.22.

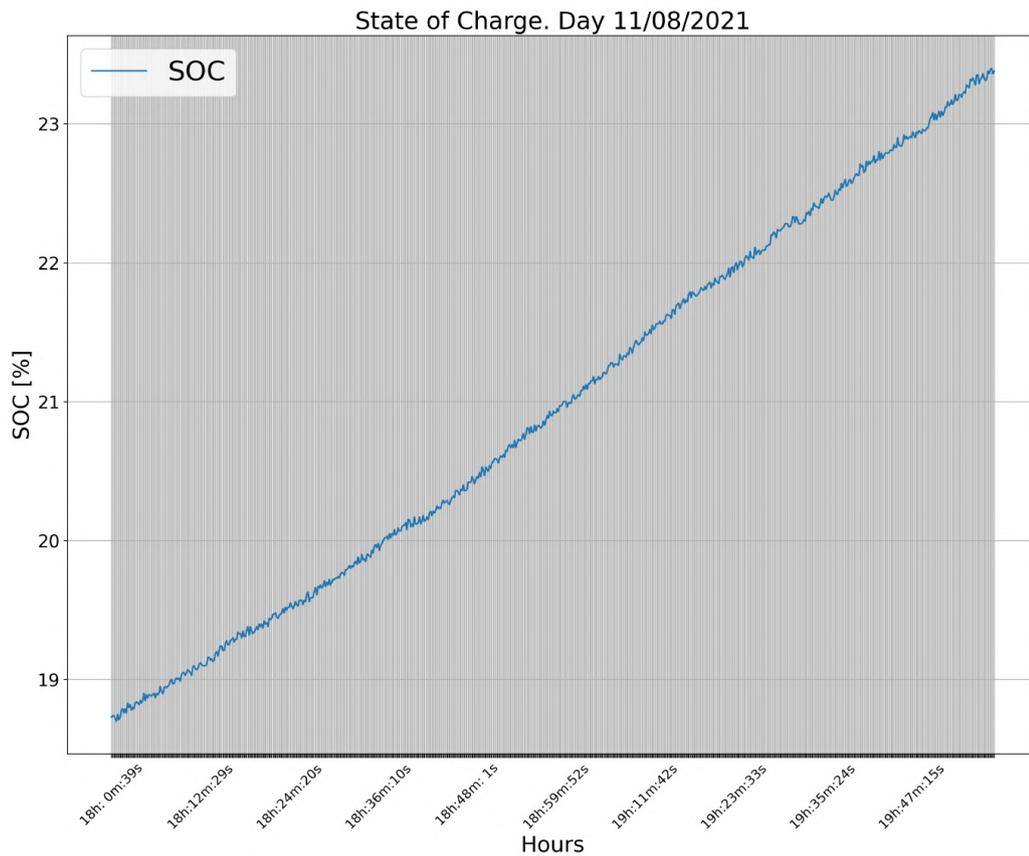


Figura 9.22: Estado de carga durante el *test* de 15A (Anexo D.20).

Finalmente, se representa en la Figura 9.23 la temperatura. Durante esta prueba, la temperatura de la batería se ha mantenido aproximadamente constante en un valor de 22.6°C.

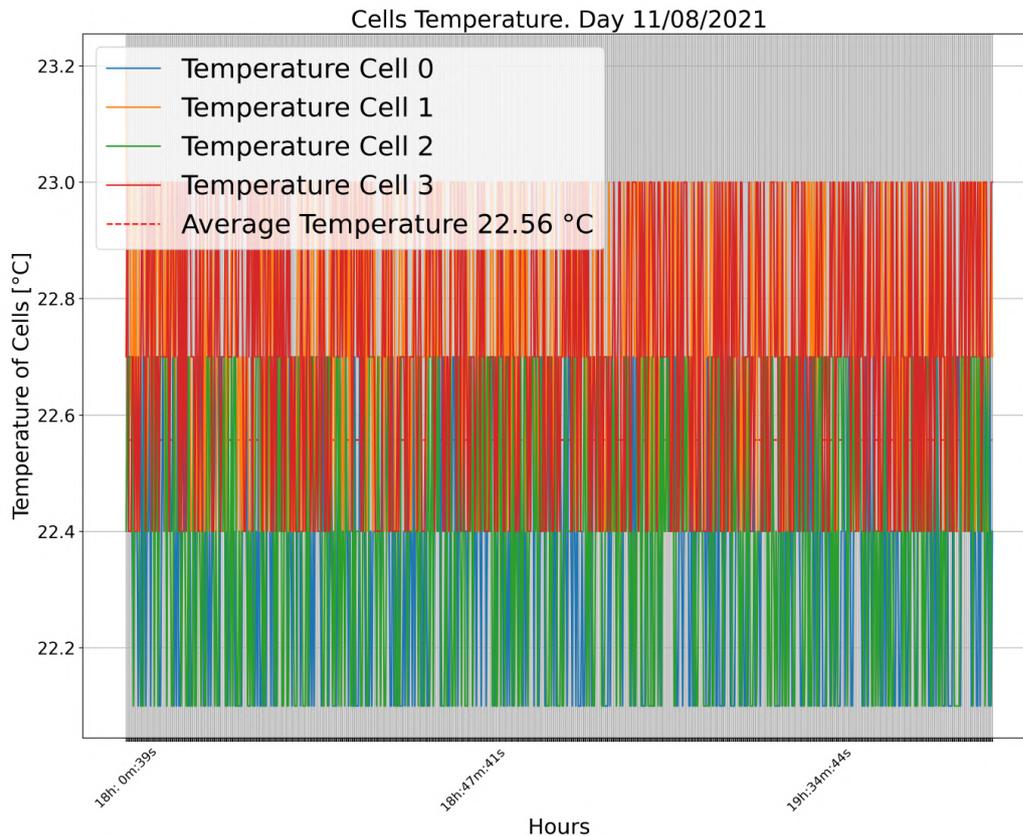


Figura 9.23: Temperaturas medidas durante el *test* de 15A (Anexo D.22).

## 9.6. Pruebas de comunicación CAN

Por otro lado, se procedió a comprobar con la ayuda de un analizador lógico de PC (modelo *SALEAE LOGIC 8*[35]) el estado del bus CAN. Para ello, se monitorizó dicho bus mientras era configurado a través de la herramienta *BMS Programmer GUI*, capturando y representando los datos de los mensajes.

### Configuración del tiempo entre mensajes

Lo primero fue configurar el tiempo que debe transcurrir entre mensajes, ajustando un valor de 220 ms. Una vez enviado y almacenado dicho valor, se procedió

a monitorizar el bus, como se muestra en la Figura 9.24, comprobando que los mensajes son enviados cada 219.29 ms.

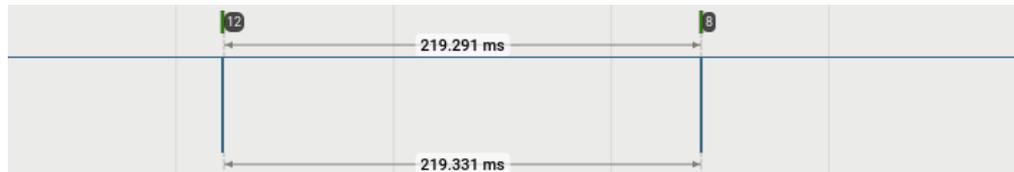


Figura 9.24: Captura con el tiempo transcurrido entre mensajes CAN.

### Pedir configuraciones

Lo siguiente que se hizo fue comprobar la comunicación para solicitar las configuraciones almacenadas. Para ello, en primer lugar se envió el comando "Z1\n" desde la interfaz, observándose el mensaje CAN con *ID 0x0B*, *DLC=1* y cuyo campo de datos contiene el mensaje esperado *0x01*, como se muestra en la Figura 9.25.

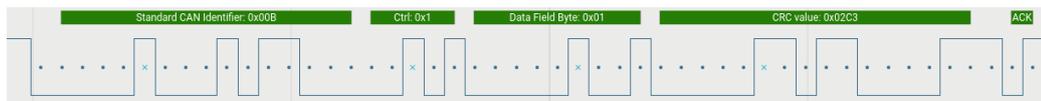


Figura 9.25: Captura con el mensaje CAN para pedir configuraciones.

Tras esto, se observa (Figura 9.26) como el BMS responde devolviendo los dos mensajes de respuesta (con identificadores *0x0C* y *0x11*) de longitud 8 y 5 bytes, respectivamente.

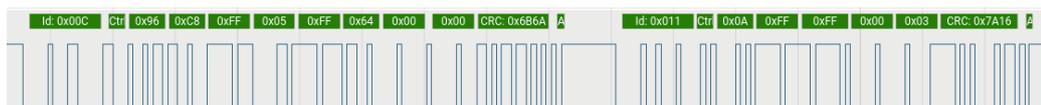


Figura 9.26: Captura con las respuestas CAN con las configuraciones.

### Configuración de algunos parámetros

Por otro lado, se ha realizado la captura de la configuración de diversos parámetros del BMS, como es el caso del valor VUV en la Figura 9.27a (*DLC=1* y *valor=0x96* que corresponde con 3.2V); VOV en la figura 9.27b (*DLC=1* y *valor=0xC8* que corresponde con 4.0 V); número de celdas en paralelo en la Figura 9.27c (*DLC=1* y *valor=0x15* que equivale a 21 celdas) o el número de termistores (Figura 9.27d con *DLC=1* y valor de *0x03*, es decir, 3 termistores).

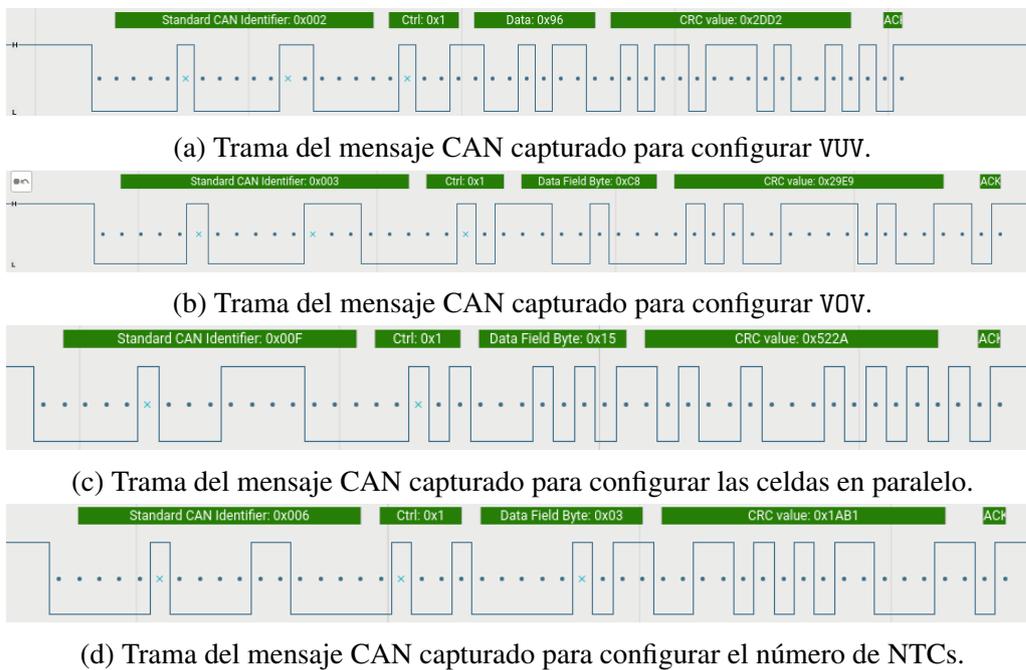


Figura 9.27: Captura de mensajes CAN transmitidos por el bus.

### Mensajes CAN de información

Finalmente, se ha realizado la captura de ciertos mensajes CAN de información de mediciones. En este caso, se muestran los mensajes de información de tensiones de las 4 primeras celdas (Figura 9.28a) y de las medidas de los primeros 8 termistores (Figura 9.28b). Para el mensaje de tensión de las celdas, los 2 bytes correspondientes a cada medición presentan los valores  $0xFF$  y  $0xF5$ , que decodificados corresponden al valor 65.525, es decir celda no conectada. En cuanto a los termistores, se reciben los 8 bytes con valores  $0x78$  que decodificado corresponde con  $34.5^{\circ}\text{C}$ .

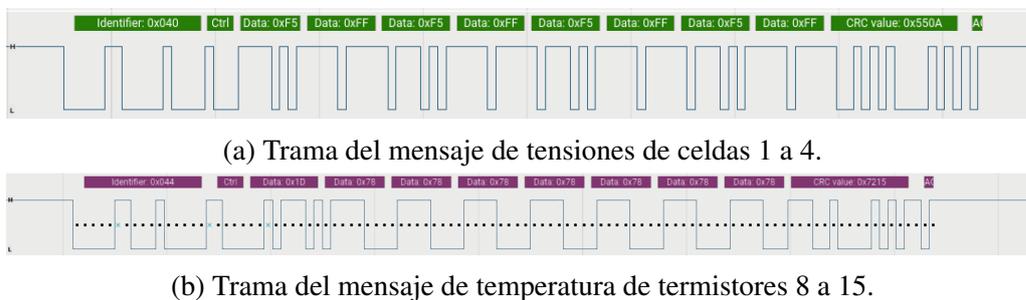


Figura 9.28: Captura de mensajes CAN transmitidos por el bus.



# Capítulo 10

## Configuración y modos de uso

En el presente capítulo se muestra el procedimiento de montaje y configuración del BMS, tanto a nivel físico como en los ajustes del *software BMS CAN Programmer*.

### 10.1. Conexionado de la batería y BMS

En la Figura 10.1 se muestra un esquema de las conexiones físicas realizadas entre la batería y el BMS. En caso de emplearse un número diferente de celdas o termistores deben conectarse de forma similar a lo expuesto. En el caso concreto de las celdas, siempre que se conecten menos de 12 celdas, todas las entradas de medición que queden sin emplear en el BMS deben conectarse a la celda con mayor potencial, de lo contrario puede provocarse un daño grave en el sistema de monitoreo. Cada conector de termistores dispone de 4 pines de *GND* con los que *alimentar* los termistores, por lo que en caso de emplearse más de cuatro, algunos de ellos deben conectarse simultáneamente a dichos pines.

En ningún caso deben dejarse terminales intermedios de medición de celdas sin conectar, o de lo contrario puede suponer un daño para el sistema BMS. Si se desea alimentar el BMS mediante una fuente externa (con una tensión superior a 11V) puede llevarse dicha alimentación a través de los pines del conector denominado *V+ EXT* y se debe modificar el *Jumper JP1* uniendo *VCC* y *EXT*.

### 10.2. Conexión del CAN2Serial y *BMS Programmer*

Para proceder a la configuración del BMS una vez este se ha conectado a la batería, debe conectarse en primer lugar el adaptador CAN2Serial al conector

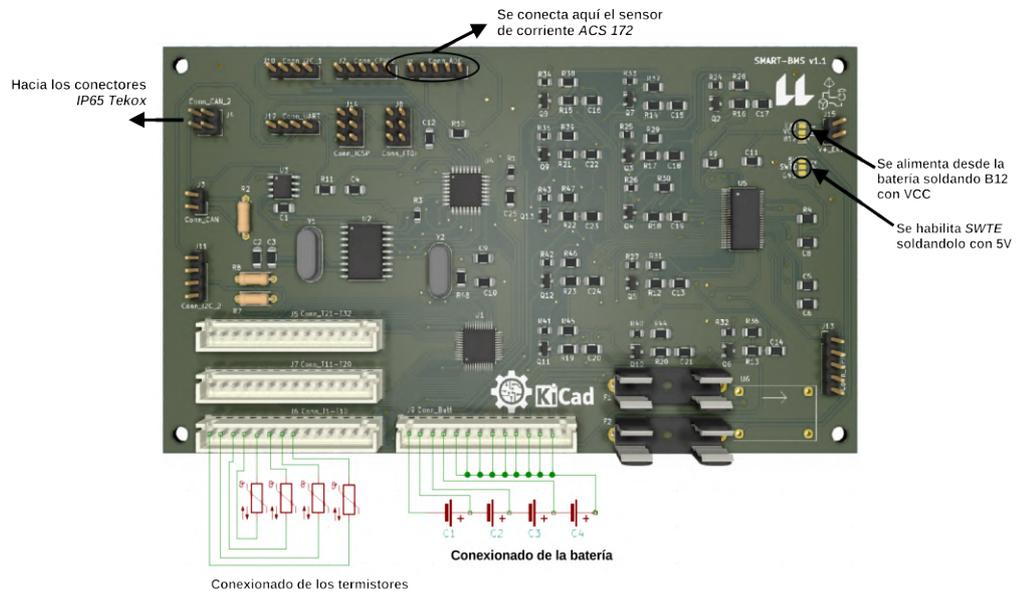


Figura 10.1: Esquema de conexión del BMS.

IP68 de la batería y a su vez este al *equipo portátil de configuración*, tal y como muestra la Figura 10.2.



Figura 10.2: Configurador y batería conectados.

Tras esto, en el *configurador* se ejecuta la interfaz *BMS Programmer GUI*, obteniéndose lo que se muestra en la Figura 10.3.



Figura 10.3: *Configurador* ejecutando el *software* de configuración.

En la pestaña *connect* se selecciona el puerto USB en el que se ha conectado la interfaz CAN2Serial y se presiona *connect*. Una vez conectado, se podrá visualizar en el *status bar* un mensaje con la última acción realizada.

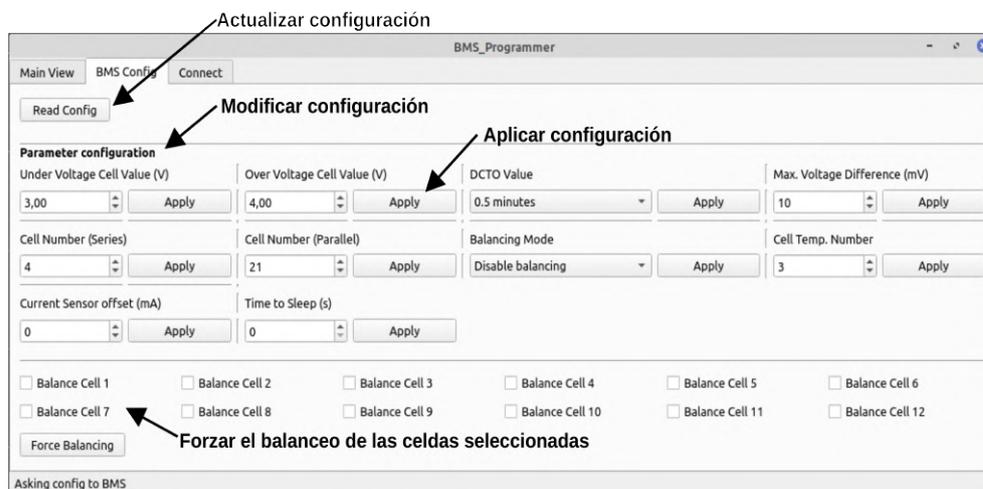


Figura 10.4: Pestaña *BMS Config* del *BMS Programmer*.

Para realizar la configuración, el usuario debe dirigirse a la pestaña *BMS Config* (Figura 10.4) y presionar en primer lugar *Read Config*. Tras esto, la interfaz se actualizará con los valores almacenados en el BMS. Una vez cargados todos los parámetros, puede procederse a modificar aquellos que el usuario desee, presionando al finalizar el botón *apply*. Si pasados 10 segundos, el parámetro modificado no se ha enviado al BMS (presionando dicho botón), se actualizará el campo mostrando nuevamente el valor almacenado. Se recomienda, tras realizar cada modificación, asegurarse que dicho parámetro ha sido recibido y actualizado en el BMS, presionando nuevamente *Read Config* y verificando que se mantiene el valor cargado.

Si el usuario desea descargar de forma intencionada una determinada celda a través de su resistencia de balanceo, debe marcar la misma en el apartado *Force Balancing* y presionar dicho botón para enviar el comando. Acto seguido, el BMS comenzará a descargar dicha celda a través de la resistencia de 2 W independientemente del valor de tensión de la misma.

Para proceder a visualizar los valores en tiempo real del estado de la batería, el usuario debe dirigirse a la pestaña *Main View* (Figura 10.5). En ella se muestran las tensiones de las celdas, indicándose en rojo aquellas que están balanceándose, las temperaturas de la batería (solo aquellas que se han configurado en *BMS Config*), el SOC, la corriente y la tensión de la batería. En caso de existir una posible advertencia en el sistema, este se mostrará en la sección *Warnings* de la interfaz.

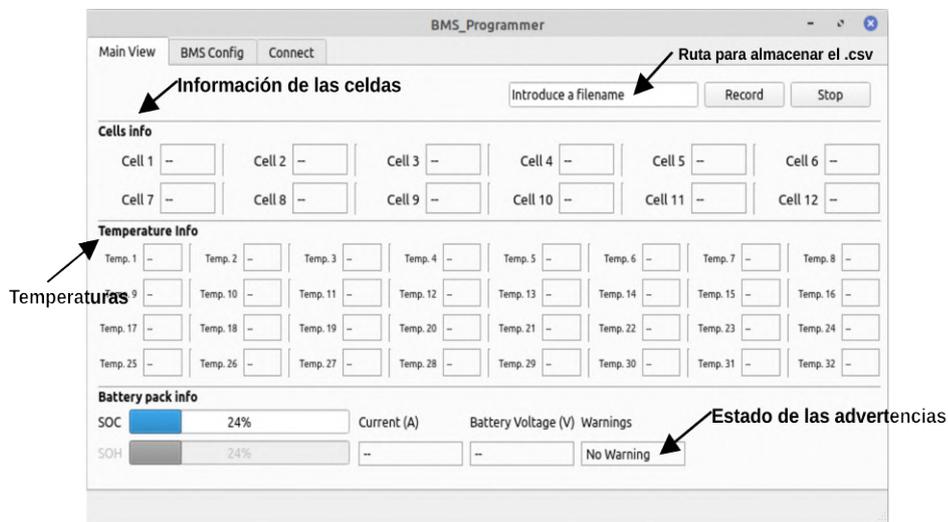


Figura 10.5: Pestaña *MainView* del *BMS Programmer*.

Finalmente, si el usuario desea capturar los datos de funcionamiento, debe dirigirse a la parte superior derecha y sustituir *Introduce a filename* por la ruta y el nombre del fichero *.csv* donde se desean almacenar los datos. Posteriormente,

debe presionar el botón *Record* para comenzar la captura y una vez finalizado, presionar *Stop*. A modo de ejemplo, si se desea almacenar los datos en un fichero llamado *datos\_20082021.csv* dentro de la carpeta Documentos, deberá introducirse la siguiente ruta:

```
/home/pi/Documentos/datos_20082021.csv
```



# Capítulo 11

## Aplicaciones y Casos de uso

En el presente capítulo se exponen otros posibles usos de la batería y el BMS diseñado, además de para alimentar sistemas aislados de captura de datos.

### 11.1. Aplicación en vehículos eléctricos

Los medios de transporte propulsados por motores eléctricos están en auge. Cada año, su cuota de mercado aumenta, estimándose que alcanzará el 22% de la cuota total dentro de cuatro años (en 2025) [3], aproximándose su precio al de los vehículos de combustión. Estos vehículos alimentan su propulsor a través de baterías (Figura 11.1), generalmente de Li-ion. Por ello, el sistema planteado puede ser de aplicación a este medio de transporte.

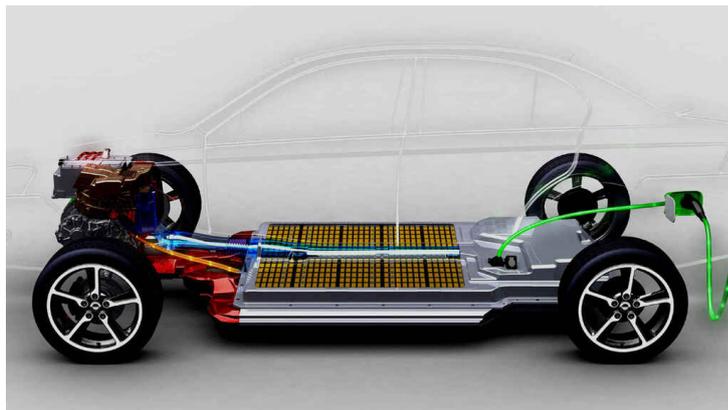


Figura 11.1: Batería en un vehículo eléctrico[55].

La tensión nominal de estas baterías suele oscilar entre los 120 V a los 400 V, con capacidades del orden de 38 kWh, en vehículos con autonomía de apro-

ximadamente 200 km. Además, estos vehículos montan una batería auxiliar de 12 V con la que alimentan todos los sistemas tradicionales del vehículo: sistema de entretenimiento, ventiladores, ordenador de a bordo, etc. Esta batería, suele presentar capacidades que oscilan entre los 45 Ah y los 85 Ah según las características del vehículo. La batería y el BMS diseñado, puede emplearse de forma directa en estos vehículos como sistema auxiliar, pudiendo emplearse la comunicación CAN del BMS para monitorizar el estado de este acumulador a través del vehículo y facilitar la carga del mismo, informando al usuario desde el sistema de control a bordo.

Debido al carácter modular del BMS diseñado, este puede emplearse para gestionar un conjunto de celdas conectadas en serie formando un grupo mayor; es decir, permite el empleo de varios BMS cada uno gestionando un *pack* de celdas que forman la batería global. De esta forma, por ejemplo, para un sistema que trabaje a tensión nominal de 120 V y emplee celdas Li-ion (con voltaje nominal de 3.7 V), es posible conformar tres paquetes de baterías de 12 celdas (el máximo admisible por el BMS), con tensión nominal de 44.4 V cada uno para formar un acumulador capaz de alimentar dicho motor eléctrico. Para ello, debe optarse por una configuración *maestro-esclavo*, es decir, uno de los BMS debe ser el dispositivo maestro que gestione las acciones de balanceo, carga, descarga, etc. asegurando así que las celdas de los diferentes paquetes se mantienen continuamente en el rango adecuado de operación (Figura 11.2).

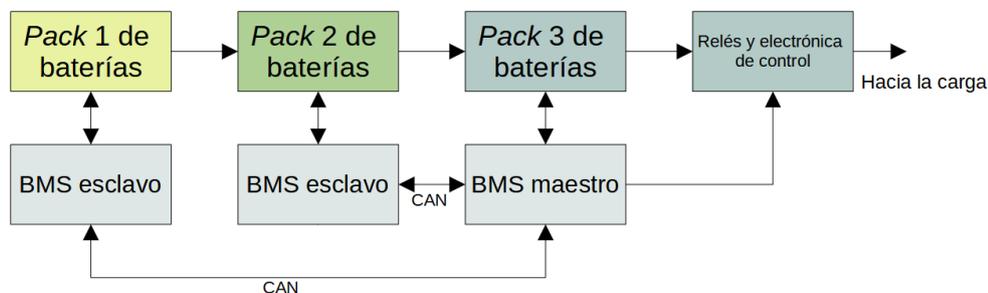


Figura 11.2: Esquema propuesto para una batería con más de 12 celdas serie.

### 11.1.1. Aplicación del prototipo al proyecto *ULL Dynamics*

La Universidad de La Laguna cuenta con un equipo de competición cuyo objetivo es el desarrollo tanto de un vehículo eléctrico de competición tipo Formula SAE, para participar en la competición **Formula Student**, como de una motocicleta eléctrica para participar en **MotoStudent** (esto último ya conseguido en Julio de 2021). En el caso del prototipo de vehículo eléctrico, emplea un sistema

de baterías de alta tensión de 400 V formado por 3 módulos 24S7P y otros 3 módulos 12S7P, lo que le proporciona una capacidad de 28Ah o 11.2 kWh. Además cuenta con un sistema de baja tensión de 24 V con el que se alimentará la electrónica de gestión del vehículo (inversor, circuitos de control, BMS de baterías de alta tensión, elementos de refrigeración, indicadores al usuario, etc), con una capacidad de 12 Ah. Este sistema auxiliar también está formado por celdas de ion-Litio Samsung INR21700 de 4 Ah de capacidad, estableciéndose así una configuración 7S3P. Se ha propuesto el uso del BMS diseñado en esta batería auxiliar, incorporando las comunicaciones al bus CAN de dicho vehículo e integrando la información de la misma en el *display* de información al piloto. Se han realizado pruebas con dicha batería, ya construida, conectada al BMS y se espera incorporar este diseño en el desarrollo del prototipo en el mes de Diciembre.



(a) Motocicleta R8-Islands construida.



(b) Diseño 3D del vehículo de *Formula Student*.

Figura 11.3: Proyectos vigentes de *ULL Dynamics*.

Por otro lado, la motocicleta construida cuenta con una batería de alta tensión formada por 16 celdas en paralelo y 30 series, formando así una batería 30S12P,

con una capacidad de 64 Ah y una tensión nominal de 111 V. Esta batería en la actualidad está gestionada por un *Orion BMS 2* [49], no obstante, se ha estudiado la aplicación del BMS diseñado en el presente proyecto a esta batería, empleando 3 módulos: 1 como *maestro* y 2 de ellos como *esclavos*. De emplearse este sistema, se garantiza una corriente de balanceo superior a la que proporciona el BMS de Orion (370 mA frente a 200 mA) [49], necesitando por tanto, un tiempo inferior para el balanceo de las celdas durante la carga. Al igual que el BMS puede aplicarse a este prototipo, el diseño modular de la batería propuesta permite adaptarse perfectamente a los requerimientos de la motocicleta, pudiendo construirse este acumulador de Alta Tensión empleando 30 *módulos básicos* de los diseñados, conectados cada uno con el siguiente en serie.

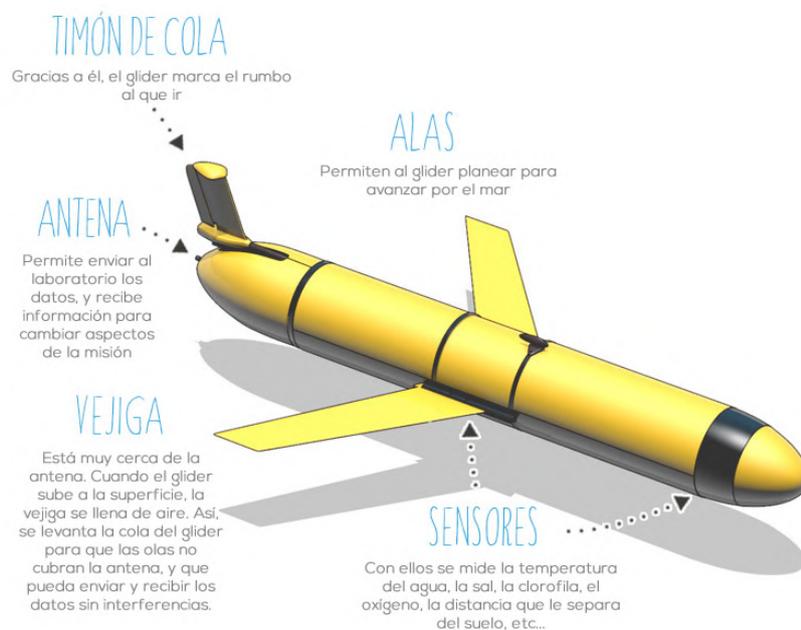


Figura 11.4: Estructura de un *glider*[48].

## 11.2. Aplicación en vehículos no tripulados

Otro posible uso de este sistema de batería y BMS radica en la alimentación de pequeños dispositivos no tripulados como *drones marinos* o *gliders* (Figura 11.4). Estos dispositivos presentan una estructura tubular alargada, con dos alas laterales y un timón de cola (con una forma muy similar a un aeroplano) y se propulsan mediante una vejiga que se llena de aire al subir a la superficie y posteriormente

lo expulsa. De esta forma el dispositivo avanza realizando a su vez movimientos verticales de ascenso y descenso. En estos dispositivos resulta de utilidad el sistema diseñado, al dotarlo de alimentación e integrar una red de comunicación que puede servir para establecer estrategias avanzadas de captura de datos basadas en la cantidad de energía restante en el sistema, conectando o desconectando de forma *inteligente* determinados sensores.



# Conclusiones

Como conclusión general, en el presente proyecto se han cumplido los siguientes objetivos:

1. Se ha diseñado y fabricado una placa electrónica capaz de monitorizar los diferentes parámetros de las celdas de la batería, realizar un equilibrado/-balanceo y protección de las celdas así como establecer una comunicación CAN BUS con el resto de equipos del sistema.
2. Se ha dimensionado una instalación solar generadora para abastecer correctamente al sistema acumulador y el de monitoreo acústico.
3. Se ha diseñado y fabricado un sistema acumulador basado en celdas de Litio para abastecer al sistema autónomo de monitorización acústica.
4. Se ha diseñado y fabricado una estructura mecánica capaz de proteger el sistema de baterías.
5. Se ha fabricado un equipo portátil para el diagnóstico del sistema.
6. Se ha diseñado una interfaz de programación para permitir la configuración y diagnóstico del BMS.
7. Se han diseñado las herramientas *software* para la correcta programación, diagnóstico y monitoreo del controlador y la batería.

Por tanto, como puede deducirse de la relación anterior, se han cumplido todos los objetivos propuestos al comienzo del presente proyecto. El diseño de un BMS y un sistema de batería modular presenta un gran potencial al permitir alimentar sistemas electrónicos muy diversos, desde pequeños equipos autónomos hasta la electrónica de vehículos de transporte y competición. La principal ventaja de este sistema radica en su fácil mantenimiento, ya que a diferencia de los sistemas de baterías convencionales, cuando una celda o grupo de celdas resulten dañados, el sistema puede ser abierto y puede sustituirse el paquete dañado, manteniendo el resto del sistema. Por otro lado, al emplearse un Sistema de Gestión

con comunicación, es posible que el resto de equipos que forman el sistema estén informados en todo momento del estado de la alimentación, algo muy útil para sistemas autónomos autogestionados, ya que de esta forma estos equipos pueden pasar a *estados de bajo consumo* o llegado el caso, apagarse hasta que se eleve nuevamente el estado de carga. Por otro lado, gracias al uso de las tecnologías de impresión 3D ha sido posible reducir el coste de la fabricación de los elementos estructurales de la batería.

## **Líneas abiertas**

Durante el desarrollo del proyecto se han encontrado una serie de posibles líneas abiertas sobre las que continuar su desarrollo. La principal es la posibilidad de desarrollar un sistema de unión de celdas más sencillo, basado en un apriete a presión, de modo que se evite el empleo de la soldadura de punto para la unión *celda-níquel* y el estaño en la unión *níquel-cobre*. Por otro lado, el empleo de este tipo de celdas 21700 permite el diseño de sistemas de baterías de diferentes formas, pudiendo adaptar el acumulador al espacio disponible o incluso distribuirlos alrededor de una determinada superficie. No obstante, estos desarrollos pueden ser abordados en futuros trabajos de fin de grado o de máster en Ingeniería.

# Conclusions

As a general conclusion, the following objectives have been achieved in this project:

1. An electronic board has been designed and manufactured, capable of monitoring the different parameters of the battery cells, balancing/balancing and protection of the cells as well as establishing CAN BUS communication with the rest of the equipment in the system.
2. A generator system has been dimensioned to supply the storage and acoustic monitoring system correctly.
3. An accumulator system based on Lithium cells has been designed and manufactured to supply the autonomous acoustic monitoring system.
4. A mechanical structure capable of protecting the battery system has been designed and manufactured.
5. A portable equipment for system diagnostics has been produced.
6. A programming interface has been designed to allow configuration and diagnostics of the BMS.
7. Software tools have been designed for the correct programming, diagnosis and monitoring of the controller and the battery.

Therefore, as can be deduced from the above, all the objectives proposed at the beginning of this project have been met. The design of a BMS and a modular battery system has great potential for powering a wide range of electronic systems, from small autonomous equipment to the electronics of transport and competition vehicles. The main advantage of this system lies in its ease of maintenance, since unlike conventional battery systems, when a cell or group of cells is damaged, the system can be opened and the damaged pack can be replaced, maintaining the rest of the system. On the other hand, by using a Management System with communication, it is possible for the rest of the equipment in the system to be

informed at all times of the state of the power supply, something very useful for self-managed autonomous systems, as in this way this equipment can switch to low consumption states or, if necessary, shut down until the state of charge is raised. On the other hand, thanks to the use of 3D printing technologies, it has been possible to reduce the cost of manufacturing the structural elements of the battery.

## **Open lines**

During the development of the project, a number of possible avenues for further development have been identified. The main one is the possibility of developing a simpler cell joining system, based on pressure tightening, so as to avoid the use of spot welding for the *cell-nickel* junction and tin for the *nickel-copper* junction. On the other hand, the use of this type of 21700 cells allows the design of battery systems of different shapes, being able to adapt the accumulator to the available space or even to distribute them around a certain surface. However, these developments can be addressed in future final degree or master's degree projects in engineering.

# Bibliografía

- [1] *¿Qué es JTAG y como puedo usarlo?* XJTAG. URL: [https://www.xjtag.com/wp-content/uploads/xjtag-ebook\\_what-is-jtag-es.pdf](https://www.xjtag.com/wp-content/uploads/xjtag-ebook_what-is-jtag-es.pdf).
- [2] *ADG731 datasheet*. URL: [https://www.analog.com/media/en/technical-documentation/data-sheets/ADG725\\_731.pdf](https://www.analog.com/media/en/technical-documentation/data-sheets/ADG725_731.pdf).
- [3] «Los coches eléctricos costarán como los de gasolina en 2027». 2021. URL: <https://motor.elpais.com/actualidad/coches-electricos-precio-baratos/>.
- [4] Atmel. «Microchip». URL: <http://www.microchip.com>.
- [5] Varios autores. «Arquitectura de microprocesadores». URL: <http://sistdig.wikidot.com/wiki:arquitectura>.
- [6] Varios autores. «Atom». URL: <http://www.atom.io>.
- [7] Varios autores. «C++». URL: <http://www.cplusplus.com>.
- [8] Varios autores. «Debian». URL: <http://www.debian.org>.
- [9] Varios autores. «FlatCAM». URL: <http://flatcam.org/>.
- [10] Varios autores. «FreeCAD». URL: <http://freecadweb.org/>.
- [11] Varios autores. «Inkscape». URL: <http://inkscape.org/>.
- [12] Varios autores. «LaTeX Reference». URL: <http://maths.cnam.fr/Membres/%20wilk/MathMax/help/LaTeX/latex.html>.
- [13] Varios autores. «Linux Mint». URL: <http://linuxmint.com>.
- [14] Varios autores. «LoRa». URL: <https://loira-alliance.org/>.
- [15] Varios autores. «Platformio». URL: <http://www.platformio.org>.
- [16] Varios autores. «Prusa Slic3r». URL: <https://www.prusa3d.es/prusaslicer/>.
- [17] Varios autores. «PVGIS». URL: <https://ec.europa.eu/jrc/en/pvgis>.
- [18] Varios autores. «PyQt». URL: <http://wiki.python.org/moin/PyQt>.
- [19] Varios autores. «Python». URL: <https://www.python.org>.

- [20] Varios autores. «Qt». URL: <http://www.qt.io>.
- [21] Varios autores. «Qt Designer». URL: <https://doc.qt.io/qt-5/qtdesigner-manual.html>.
- [22] Varios autores. «TeXstudio». URL: <http://www.texstudio.org>.
- [23] Varios autores. «Thingiverse». URL: [www.thingiverse.com](http://www.thingiverse.com).
- [24] AutoWp. «mcp2515.h». URL: <https://github.com/autowp/arduino-mcp2515>.
- [25] *BSS308PE Datasheet*. URL: [https://www.infineon.com/dgdl/Infineon-BSS308PE-DS-v02\\_03-en.pdf?fileId=db3a304330f686060131099c80400073](https://www.infineon.com/dgdl/Infineon-BSS308PE-DS-v02_03-en.pdf?fileId=db3a304330f686060131099c80400073).
- [26] Gobierno de Canarias. «GRAFCAN». URL: <https://visor.grafcan.es/visorweb/>.
- [27] J Cao y A Emadi. «Batteries Need Electronics». En: *IEEE Industrial Electronics Magazine* 5.1 (2011), pág. 9. ISSN: 1932-4529. DOI: [10.1109/mie.2011.940251](https://doi.org/10.1109/mie.2011.940251).
- [28] Linus Torvalds y community. «Git: a Free and Open Source Distributed Version Control System». URL: <https://git-scm.com/>.
- [29] P. Deja. «Tests of BMS Battery Management System with active and passive system of balancing the battery capacity». En: *IOP Conference Series: Materials Science and Engineering*. Vol. 679. 1. 2019. DOI: [10.1088/1757-899X/679/1/012009](https://doi.org/10.1088/1757-899X/679/1/012009). URL: <https://iopscience.iop.org/article/10.1088/1757-899X/679/1/012009>.
- [30] EPEver. *EPSolar 3210AN datasheet*.
- [31] *Hidrófono AS-1 Datasheet*. Aquarian. URL: <https://www.aquarianaudio.com/as-1-hydrophone.html>.
- [32] *Hifi Berry DAC+ADC Datasheet*. URL: <https://www.hifiberry.com/docs/data-sheets/datasheet-dac-adc/>.
- [33] IDAE. *Pliego de Condiciones Técnicas de Instalaciones Aisladas de Red*.
- [34] *INA 219 Datasheet*. Texas Instrument. URL: <https://www.ti.com/lit/ds/symlink/ina219.pdf>.
- [35] Saleae Inc. «Saleae Logic 8». URL: <https://www.saleae.com/es/>.
- [36] *Introduction to SPI Interface*. Analog Dialogue. URL: <https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html>.
- [37] *Introduction to the Controller Area Network (CAN)*. Texas Instrument. URL: <https://www.ti.com/lit/an/sloa101b/sloa101b.pdf?ts=1630184126010>.

- [38] Universidad de Jaen. «Calculadora de la radiación media mensual». URL: [http://www.ujaen.es/investiga/solar/07cursosolar/home\\_main\\_frame/02\\_radiacion/01\\_basico/java/pro\\_rad\\_5.htm](http://www.ujaen.es/investiga/solar/07cursosolar/home_main_frame/02_radiacion/01_basico/java/pro_rad_5.htm).
- [39] «JLCPCB». URL: <https://www.jlpcb.com>.
- [40] Sigi C. Joseph y col. «Battery Management System for DC Nanogrid». En: *2018 20th National Power Systems Conference, NPSC 2018*. 2018, pág. 5. DOI: [10.1109/NPSC.2018.8771838](https://doi.org/10.1109/NPSC.2018.8771838).
- [41] *Juice4Halt UPS Datasheet*. NElectra. URL: <https://juice4halt.com/wp-content/uploads/2017/11/datasheet20-5V.pdf>.
- [42] Dan Kehoe. «7Raspberry Pi display Case». URL: <https://www.thingiverse.com/thing:1803757>.
- [43] Markus Lelie y col. «Battery management system hardware concepts: An overview». En: *Applied Sciences (Switzerland)* 8.4 (2018). ISSN: 20763417. DOI: [10.3390/app8040534](https://doi.org/10.3390/app8040534).
- [44] Kailong Liu y col. «A brief review on key technologies in the battery management system of electric vehicles». En: *Frontiers of Mechanical Engineering* 14.1 (2019). ISSN: 20950241. DOI: [10.1007/s11465-018-0516-8](https://doi.org/10.1007/s11465-018-0516-8).
- [45] *LTC6804 Datasheet*. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/680412fc.pdf>.
- [46] Augusto Samuel Hernández Martín. «Repositorio BMS-BatterySystem». URL: <https://github.com/AugustoS97/BMS-BatterySystem.git>.
- [47] Vicente Mascarós Mateo. *Instalaciones generadoras fotovoltaicas*. Ed. por Paraninfo. 2015.
- [48] Medclíc. «Así es por dentro un glider». URL: <https://medclíc.es/es/instrumentos/glider/>.
- [49] «Orion BMS». URL: <https://www.orionbms.com/products/orion-bms-standard/>.
- [50] *Protocolo MODBUS*. National Instrument. URL: <https://www.ni.com/es-es/innovations/white-papers/14/the-modbus-protocol-in-depth.html>.
- [51] *Raspberry Pi 7" Touch Screen Display*. URL: <https://docs.rs-online.com/75d2/A700000007993904.pdf>.
- [52] Rigol. «DL3021». URL: <https://www.rigolna.com/products/dc-power-loads/dl3000/>.

- [53] *Samsung INR21700-40T Datasheet*. URL: <https://www.dnkpower.com/wp-content/uploads/2019/02/SAMSUNG-INR21700-40T-Datasheet.pdf>.
- [54] The KiCad Team. «KiCad EDA: A Cross Platform and Suite, Open Source Electronics Design Automation». URL: <http://docs.kicad-%20pcb.org/es/kicad.html>.
- [55] «Tipos de batería de un coche eléctrico». 2021. URL: <https://www.lovesharing.com/tipos-de-bateria-de-un-coche-electrico/>.



**Escuela de Doctorado  
y Estudios de Posgrado**  
Universidad de La Laguna

ESCUELA DE ESTUDIOS DE  
DOCTORADO Y POSTGRADO

## **Trabajo de Fin de Master**

**Desarrollo de un BMS y su batería para sistemas  
aislados**

### **TOMO II**

**Pliego de condiciones y presupuesto**

**Titulación:** Máster en Ingeniería Industrial

**Estudiante:**

Augusto Samuel Hernández Martín

**Tutor:** Fernando Luis Rosa González

**Co-Tutor:** Iván Rodríguez Méndez

5 de septiembre de 2021



# Capítulo 12

## Pliego de condiciones

La principal condición para la realización de este proyecto consiste en que el desarrollo del sistema BMS debe usar *software* y *hardware* libre, y además debe ser funcional al final de su desarrollo. Otras condiciones para el desarrollo del proyecto son las siguientes:

- Emplear un sistema operativo Linux.
- Utilizar KiCad EDA para el desarrollo del diseño electrónica.
- Emplear Git para la compartición de archivos y documentos del proyecto.
- Emplear FreeCAD para el diseño de piezas mecánicas.
- Utilizar Python para el desarrollo de software.

Por otro lado, todo el *hardware* y *software* desarrollado durante el presente proyecto será libre.



# Capítulo 13

## Presupuesto del Proyecto

El presupuesto del proyecto asciende a un total de **7.032,27 €**. De los cuales **6.000,00 €** son gastos de personal y **1.032,27 €** son gastos de material y de fabricación. En la Tabla 13.1, se descompone dicho presupuesto, donde se indica los elementos implicados, su cantidad, su precio unitario (si procede) y el precio total.

Elemento	Cantidad (ud)	Precio (€/ud)	Total(€)
Componentes electrónicos	180	-	195,91 €
Placas de circuito impreso	5	-	54,90 €
Componentes mecánicos	831	-	46,40 €
Bobinas para impresora 3D	4	21,50 €	86,00 €
Celdas de la batería	84	6,34 €	532,56 €
Componentes eléctricos	45	-	116,50 €
Precio por hora de diseño mecánico	100	20,00 €	2.000,00 €
Precio por hora de diseño electrónico	100	20,00 €	2.000,00 €
Precio por hora de diseño de software	100	20,00 €	2.000,00 €
		<b>TOTAL</b>	<b>7.032,27 €</b>

Tabla 13.1: Descomposición del presupuesto general del proyecto

Para la fabricación de la mecánica y las placas de circuito impreso, únicamente se ha tenido en cuenta el coste de material, no incluyéndose coste de fabricación. Por otro lado, se ha establecido un precio de 20€/h para el trabajo de diseño mecánico, electrónico y de *software*.

El desglose del coste de los componentes electrónicos empleados se puede ver en la Tabla 13.2.

Grupo	Componente	Distribuidor	Unidades	€/ud	€
Condensadores	Condensador 100nF SMD 1206	LCSC	11	0,07 €	0,77 €
Condensadores	Condensador 22pF SMD 1206	LCSC	6	0,06 €	0,36 €
Condensadores	Condensador 27pF SMD 1206	LCSC	6	0,09 €	0,54 €
Condensadores	Condensador 10nF SMD 1206	LCSC	12	0,07 €	0,84 €
Condensadores	Condensador 1uF SMD 1206	LCSC	10	0,11 €	1,10 €
Fusible	Fusible 5x20mm 1A	LCSC	3	0,32 €	0,96 €
Portafusible	Portafusible 5x20mm 1A	LCSC	3	1,20 €	3,60 €
Conectores	Conector PinHeader	LCSC	12	1,78 €	21,36 €
Conectores	Conector JST XH 1x14 2.54 mm	LCSC	4	3,48 €	13,92 €
Conectores	Conector IDC 2x05 2.54mm	LCSC	1	2,50 €	2,50 €
Conectores	Conector Bloque Terminal 2x5.08 mm	LCSC	7	0,25 €	1,75 €
Conectores	Conector RJ45	LCSC	1	0,90 €	0,90 €
Transistores	BSS308PE SMD SOT23	LCSC	12	0,39 €	4,68 €
Multiplexor	Multiplexor ADG731 TQFP-48	RS	1	9,58 €	9,58 €
Integrados CAN	MCP2515 SMD SOIC18	LCSC	3	2,07 €	6,21 €
Integrados CAN	Transceiver TJA1050T	LCSC	3	1,20 €	3,60 €
Microcontrolador	ATMega 328P TQFP-32	LCSC	3	0,95 €	2,85 €
Microcontrolador	Arduino Nano	LCSC	1	3,20 €	3,20 €
Integrado BMS	LTC6804 SSOP-48	MOUSER	1	18,02 €	18,02 €
Integrado serial	MAX232 DIP16	NALBER	1	1,20 €	1,20 €
Integrado serial	MAX485 DIP8	NALBER	1	2,77 €	2,77 €
Cuarzo	Cristal 8MHz HC49	LCSC	3	0,31 €	0,93 €
Cuarzo	Cristal 16MHz HC49	LCSC	3	0,31 €	0,93 €
Resistencia	Resistencia 10kOhm 0805	RS	11	0,03 €	0,33 €
Resistencia	Resistencia 1kOhm 0805	RS	3	0,03 €	0,09 €
Resistencia	Resistencia 120Ohm SMD 1206	LCSC	17	0,10 €	1,70 €
Resistencia	Resistencia 100Ohm SMD 1206	LCSC	13	0,10 €	1,30 €
Resistencia	Resistencia 1Mhm SMD 1206	LCSC	3	0,10 €	0,30 €
Resistencia	Resistencia 100Ohm 2W SMD 1206	MOUSER	12	4,90 €	58,80 €
Resistencia	Resistencia 220Ohm THD	LCSC	3	0,10 €	0,30 €
Resistencia	Resistencia 20kOhm SMD 1206	LCSC	3	0,10 €	0,30 €
Termistor	NTC 10kOhm NTCLE300	MOUSER	12	1,20 €	14,40 €
Diodo	Diodo LED THD 3.0mm	LCSC	3	0,05 €	0,15 €
Diodo	Diodo LED SMD 1206	LCSC	1	0,07 €	0,07 €
Convertor DCDC	Convertor DCDC MP2307	NALBER	3	5,20 €	15,60 €
				<b>TOTAL</b>	<b>195,91 €</b>

Tabla 13.2: Precio detallado de los componentes electrónicos.

El coste de los elementos para la fabricación de las diferentes placas de circuito impreso (sin incluir componentes) se muestra en la Tabla 13.3.

Grupo	Descripción	Distribuidor	Unidades	€/ud	€
Circuito impreso	Placas de cobre de Circuito impreso doble cara	RS	4	5,50 €	22,00 €
Fabricación	Fabricación PCB BMS	JLC PCB	1	32,90 €	32,90 €
				<b>TOTAL</b>	<b>54,90 €</b>

Tabla 13.3: Presupuesto elementos de fabricación de PCB.

Por otro lado, el presupuesto de los elementos eléctricos para la fabricación de la batería se muestra en la Tabla 13.4.

Grupo	Componente	Distribuidor	Unidades	€/ud	€
Terminales	Terminal M4	RADIOVISIÓN	8	0,10 €	0,80 €
Terminales	Terminal M8	RADIOVISIÓN	4	0,60 €	2,40 €
Termorretractil	Termorretractil	RADIOVISIÓN	3	0,90 €	2,70 €
Conectores	Conectores TEKOX IP68 3P Macho	COELCA	3	4,90 €	14,70 €
Conectores	Conectores TEKOX IP68 3P Macho	COELCA	3	4,70 €	14,10 €
Conectores	Conectores DB9 aéreo macho	RADIOVISIÓN	4	0,90 €	3,60 €
Conectores	Conectores DB9 aéreo hembra	RADIOVISIÓN	4	0,90 €	3,60 €
Cables	Cable 4 mm <sup>2</sup> (por metro lineal) H07Z1	RADIOVISIÓN	1	2,90 €	2,90 €
Cables	Cable 1 mm <sup>2</sup> (por metro lineal)	RADIOVISIÓN	4	1,60 €	6,40 €
Cables	Cable comunicación trenzado 4 hilos (por metro lineal)	COELCA	4	1,25 €	5,00 €
Cables	Cable UTP (por metro lineal)	COELCA	2	0,60 €	1,20 €
Pletina	Pletina Níquel 0.2x8mm (rollo de 10 m)	CANARIBAT	2	17,98 €	35,96 €
Pletina	Pletina Cobre (precio por m <sup>2</sup> )	BONNET	0,018	420,00 €	7,56 €
Bornes	Borne batería	RADIOVISIÓN	2	7,79 €	15,58 €
Protector	Rollo aislante celdas 12cmx1m	RS	1	6,91 €	6,91 €
				<b>TOTAL</b>	<b>116,50 €</b>

Tabla 13.4: Precio detallado de los componentes eléctricos de la batería.

Por último, en la Tabla 13.5 se detalla el coste de los elementos necesarios para la fabricación mecánica de la batería y las diferentes carcasas protectoras.

Grupo	Componente	Distribuidor	Unidades	€/ud	€
Tornillos	Tornillo DIN7985 INOX A4 4x20	BONNET	140	0,05 €	7,28 €
Tornillos	Tornillo DIN7985 INOX A4 4x30	BONNET	30	0,08 €	2,40 €
Tornillos	Tornillo DIN7985 INOX A4 4x10	BONNET	20	0,04 €	0,72 €
Tornillos	Tornillo DIN7985 INOX A4 3x10	BONNET	20	0,05 €	1,00 €
Tornillos	Tornillo DIN7985 INOX A4 5x20	BONNET	140	0,11 €	15,40 €
Tornillos	Tornillo DIN7985 INOX A4 5x30	BONNET	30	0,12 €	3,60 €
Tornillos	Tornillo DIN7985 INOX A4 5x16	BONNET	30	0,11 €	3,30 €
Arandelas	Arandela INOX Plana A4 DIN 125 M4	BONNET	30	0,05 €	1,56 €
Arandelas	Arandela INOX Plana A4 DIN 125 M3	BONNET	30	0,01 €	0,30 €
Arandelas	Arandela INOX Plana A4 DIN 125 M5	BONNET	140	0,02 €	2,52 €
Tuercas	Tuerca cuadrada DIN 557 INOX A4 M5	BONNET	140	0,05 €	7,00 €
Tuercas	Tuerca hexagonal INOX 934 A4 M5	BONNET	40	0,02 €	0,80 €
Tuercas	Tuerca hexagonal INOX 934 A4 M4	BONNET	40	0,01 €	0,52 €
Adhesivo	Resina Epoxi Araldite 15 ml	LEROY MERLIN	1	7,90 €	7,90 €
				<b>TOTAL</b>	<b>46,40 €</b>

Tabla 13.5: Precio detallado de los elementos mecánicos.





**Escuela de Doctorado  
y Estudios de Posgrado**  
Universidad de La Laguna

ESCUELA DE ESTUDIOS DE  
DOCTORADO Y POSTGRADO

## **Trabajo de Fin de Master**

**Desarrollo de un BMS y su batería para sistemas  
aislados**

### **TOMO III**

**Anexos**

**Titulación:** Máster en Ingeniería Industrial

**Estudiante:**

Augusto Samuel Hernández Martín

**Tutor:** Fernando Luis Rosa González

**Co-Tutor:** Iván Rodríguez Méndez

5 de septiembre de 2021

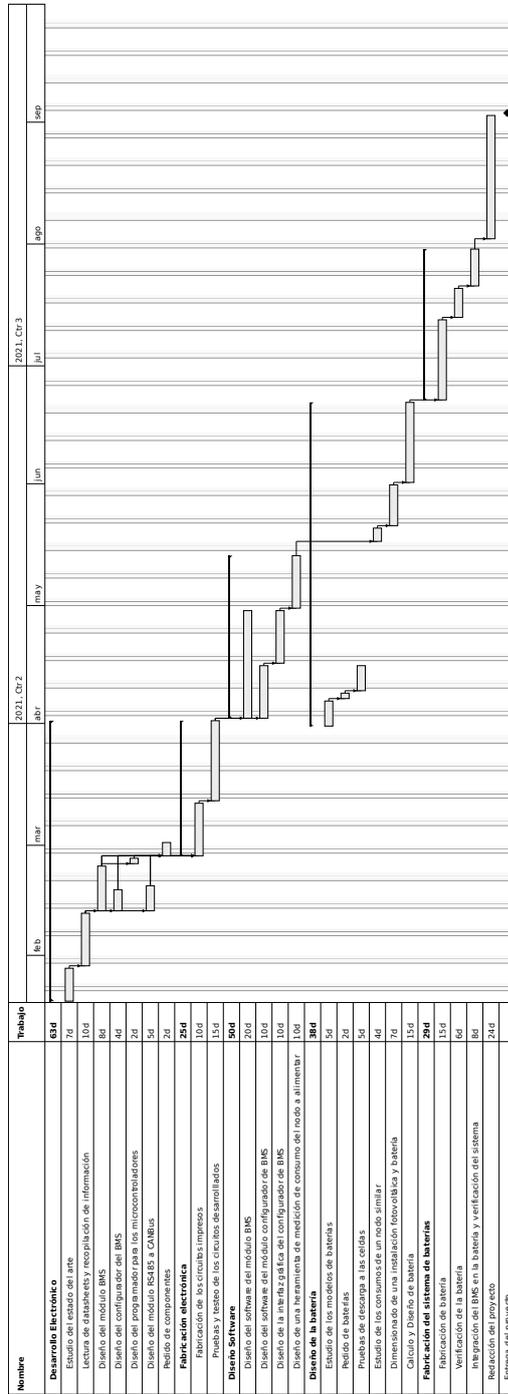


# Apéndice A

## Cronología del Proyecto

En este anexo se presenta la cronología que ha seguido el proyecto en su desarrollo. Se mostrará un diagrama de Gantt indicando las precedencias entre tareas y paquetes de trabajo, y una tabla donde se indican las fechas de inicio y fin de cada una de ellas.

# A.1. Diagrama de Gantt



## A.2. Cuadro de Temporalización

WBS	Nombre	Inicio	Fin	Trabajo	Duración	Desperdicio	Coste	Asignado a	% Completado
1	<b>Desarrollo Electrónico</b>	ene 20	abr 1	63d	52d	110d	0		0
1.1	Estudio del estado del arte	ene 20	ene 28	7d	7d		0		0
1.2	Lectura de datasheets y recopilación de información	ene 29	feb 11	10d	10d		0		0
1.3	Diseño del módulo BMS	feb 12	feb 23	8d	8d		0		0
1.4	Diseño del configurador del BMS	feb 12	feb 17	4d	4d	6d	0		0
1.5	Diseño del programador para los microcontroladores	feb 24	feb 25	2d	2d		0		0
1.6	Diseño del módulo RS485 a CANBus	feb 12	feb 18	5d	5d	5d	0		0
1.7	Pedido de componentes	feb 26	mar 1	2d	2d	133d	0		0
1.8	<b>Fabricación electrónica</b>	feb 26	abr 1	25d	25d	110d	0		0
1.8.1	Fabricación de los circuitos impresos	feb 26	mar 11	10d	10d		0		0
1.8.2	Pruebas y testeo de los circuitos desarrollados	mar 12	abr 1	15d	15d		0		0
2	<b>Diseño Software</b>	abr 2	may 13	50d	30d	80d	0		0
2.1	Diseño del software del módulo BMS	abr 2	abr 29	20d	20d	90d	0		0
2.2	Diseño del software del módulo configurador de BMS	abr 2	abr 15	10d	10d		0		0
2.3	Diseño de la interfaz gráfica del configurador de BMS	abr 16	abr 29	10d	10d		0		0
2.4	Diseño de una herramienta de medición de consumo del nodo a alimentar	abr 30	may 13	10d	10d		0		0
3	<b>Diseño de la batería</b>	mar 31	jun 21	38d	59d	53d	0		0
3.1	Estudio de los modelos de baterías	mar 31	abr 6	5d	5d	100d	0		0
3.2	Pedido de baterías	abr 7	abr 8	2d	2d	100d	0		0
3.3	Pruebas de descarga a las celdas	abr 9	abr 15	5d	5d	100d	0		0
3.4	Estudio de los consumos de un nodo similar	may 17	may 20	4d	4d		0		0
3.5	Dimensionado de una instalación fotovoltaica y batería	may 21	may 31	7d	7d		0		0
3.6	Calculo y Diseño de batería	jun 1	jun 21	15d	15d		0		0
4	<b>Fabricación del sistema de baterías</b>	jun 22	jul 30	29d	29d	24d	0		0
4.1	Fabricación de batería	jun 22	jul 12	15d	15d		0		0
4.2	Verificación de la batería	jul 13	jul 20	6d	6d		0		0
4.3	Integración del BMS en la batería y verificación del sistema	jul 21	jul 30	8d	8d		0		0
5	Redacción del proyecto	ago 2	sep 2	24d	24d		0		0
6	Entrega del proyecto	sep 3	sep 3	N/D	N/D		0		0



## **Apéndice B**

### **Gráficos caracterización de las celdas**

A continuación se presentan las gráficas de caracterización de las celdas, obtenidas a partir del ensayo de descarga expuesto en el Capítulo 6.

### B.1. Curva de SOC de la celda

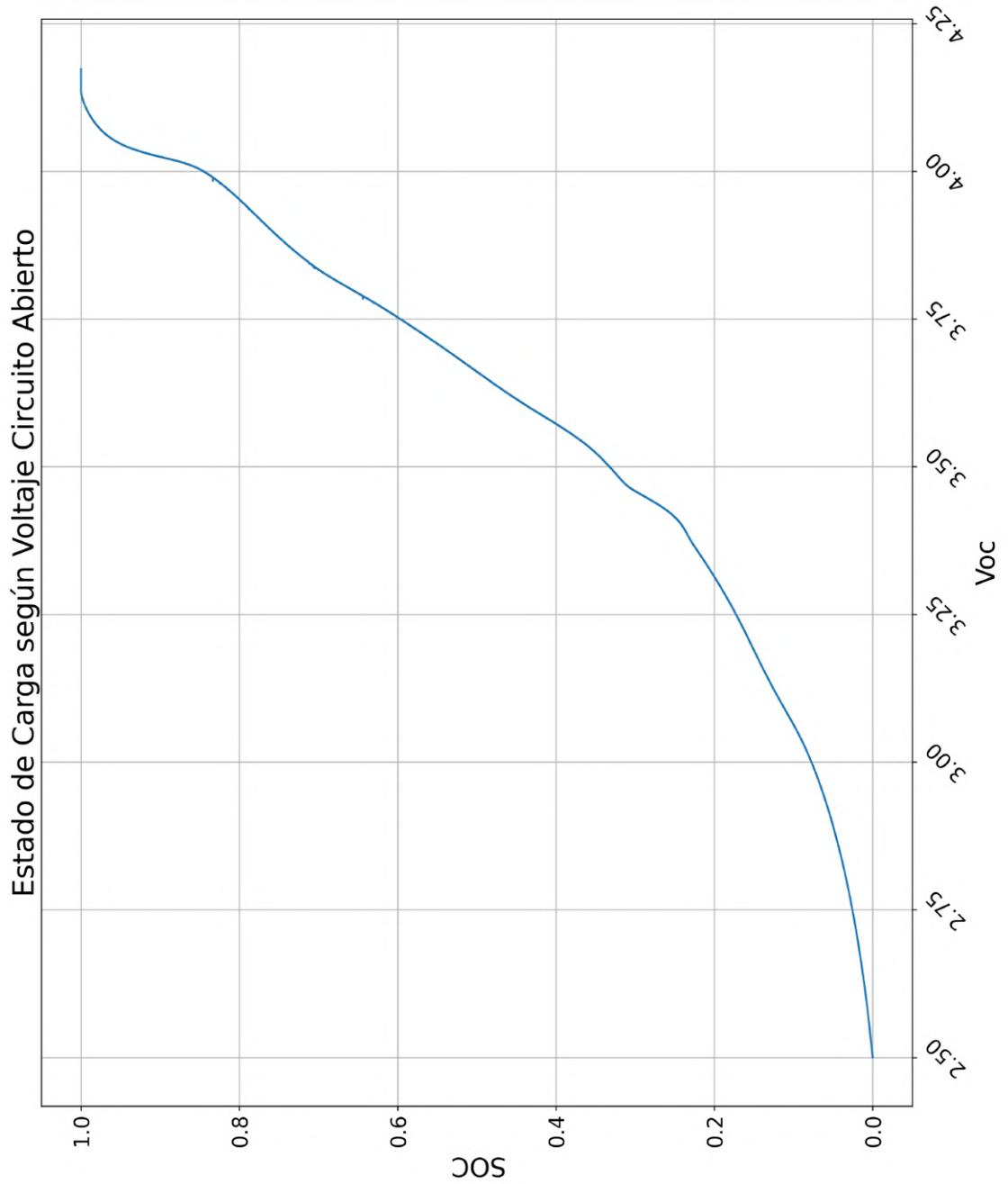


Figura B.1: Curva generada tras el ensayo de descarga de la celda.

## B.2. Regresión polinómica de la curva SOC

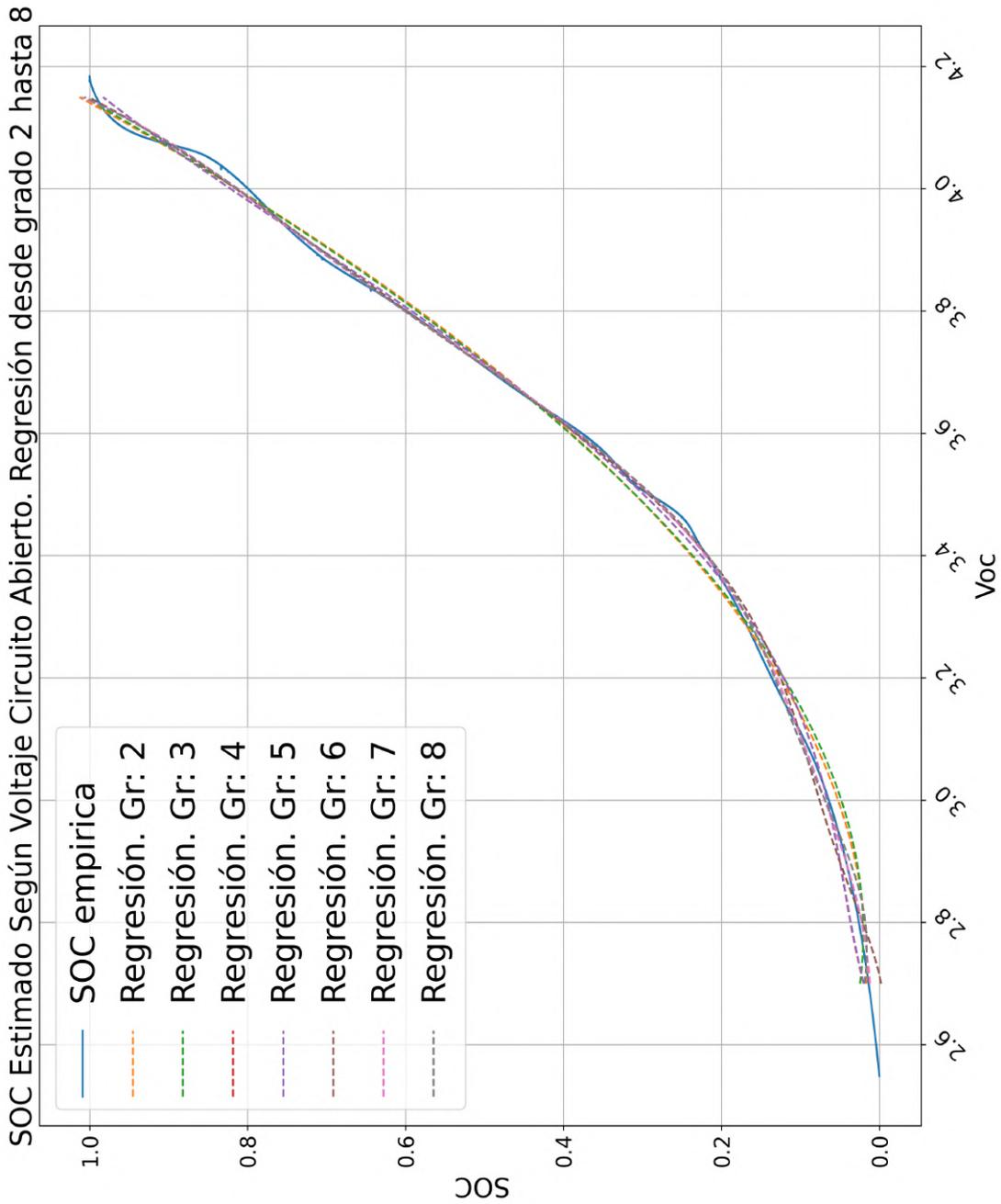


Figura B.2: Conjunto de curvas de regresión polinómica generadas.

### B.3. Regresión polinómica de la curva SOC. Grado 2

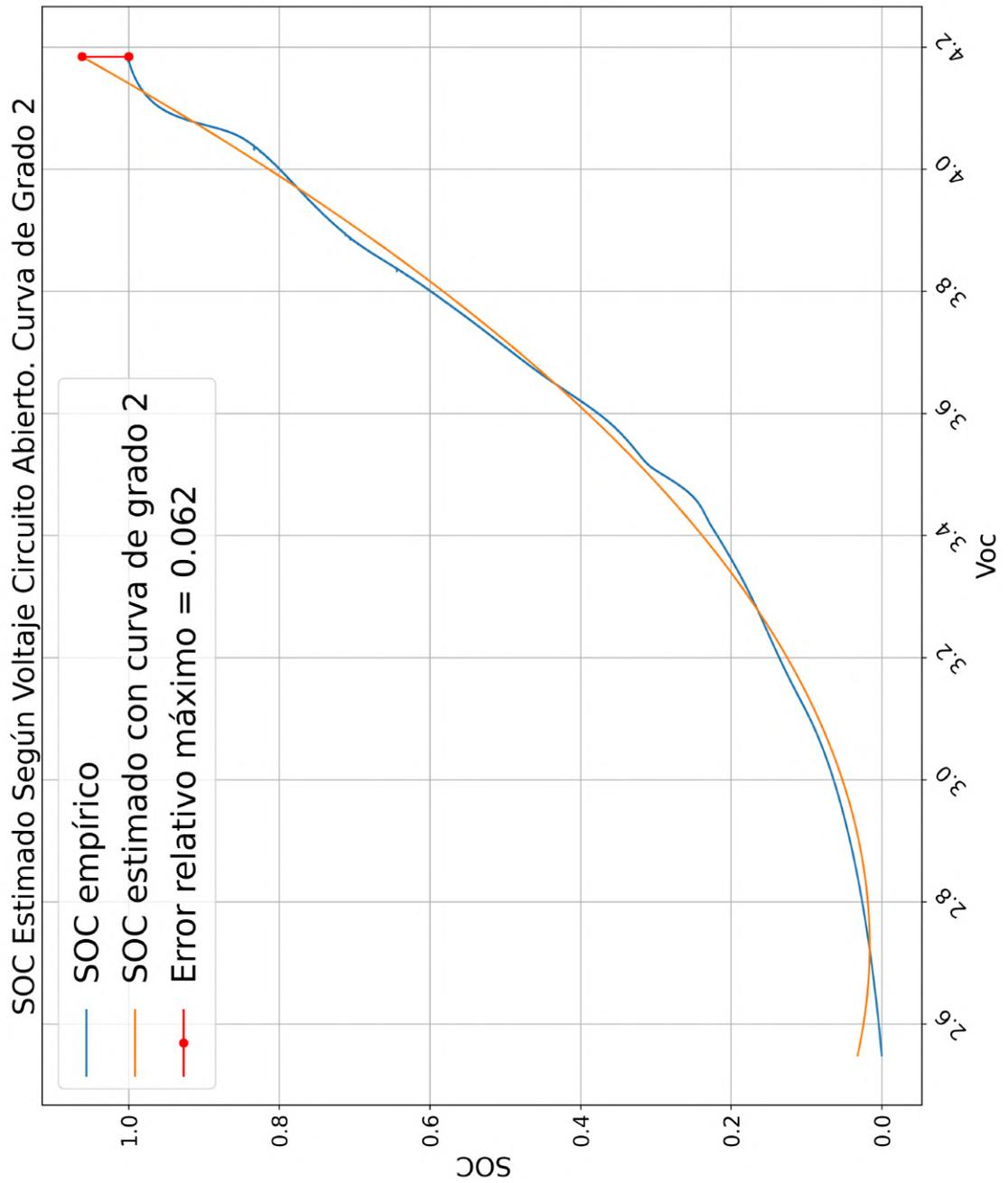


Figura B.3: Curva de regresión del SOC de grado 2.

### B.4. Regresión polinómica de la curva SOC. Grado 3

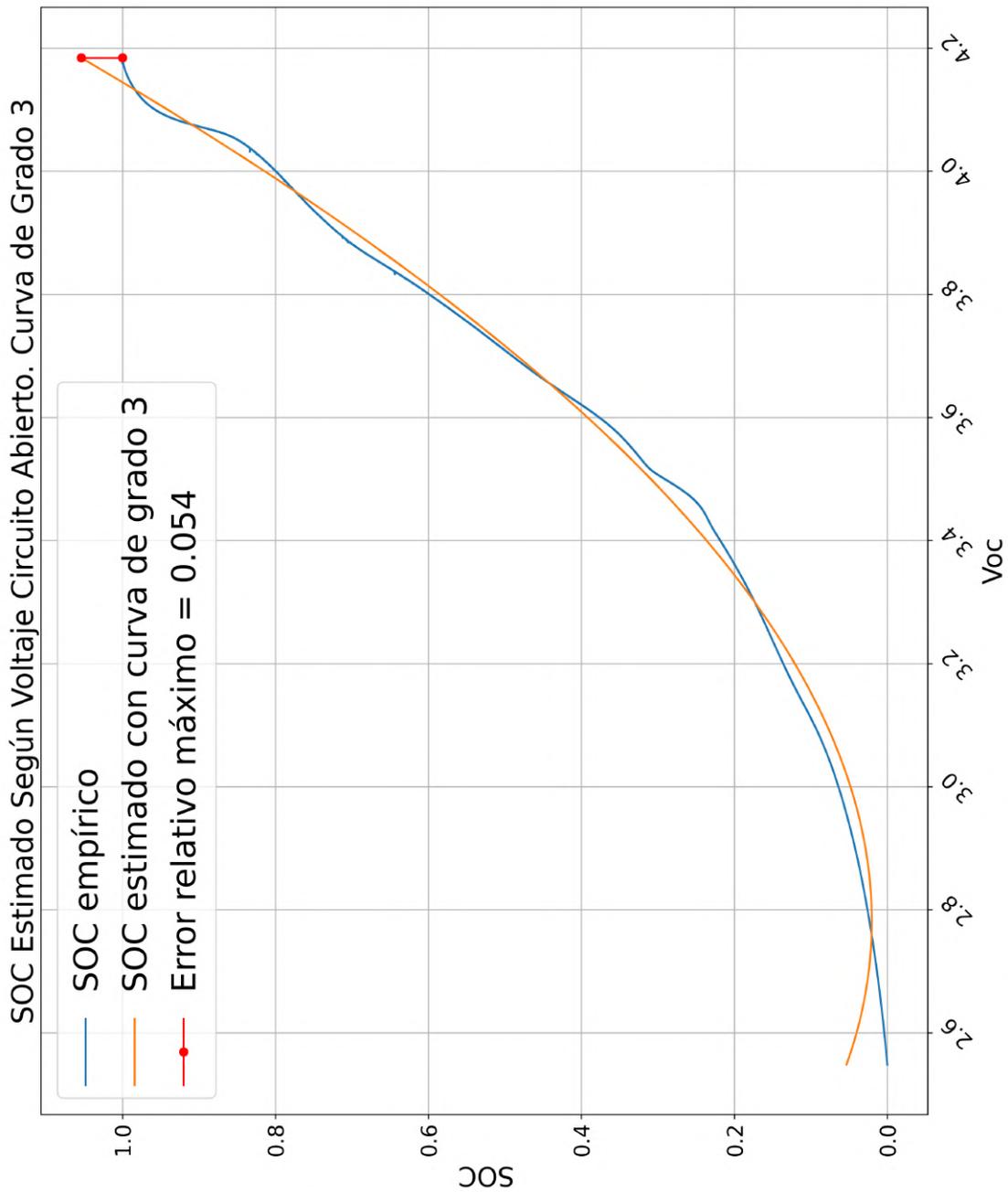


Figura B.4: Curva de regresión del SOC de grado 3.

## B.5. Regresión polinómica de la curva SOC. Grado 4

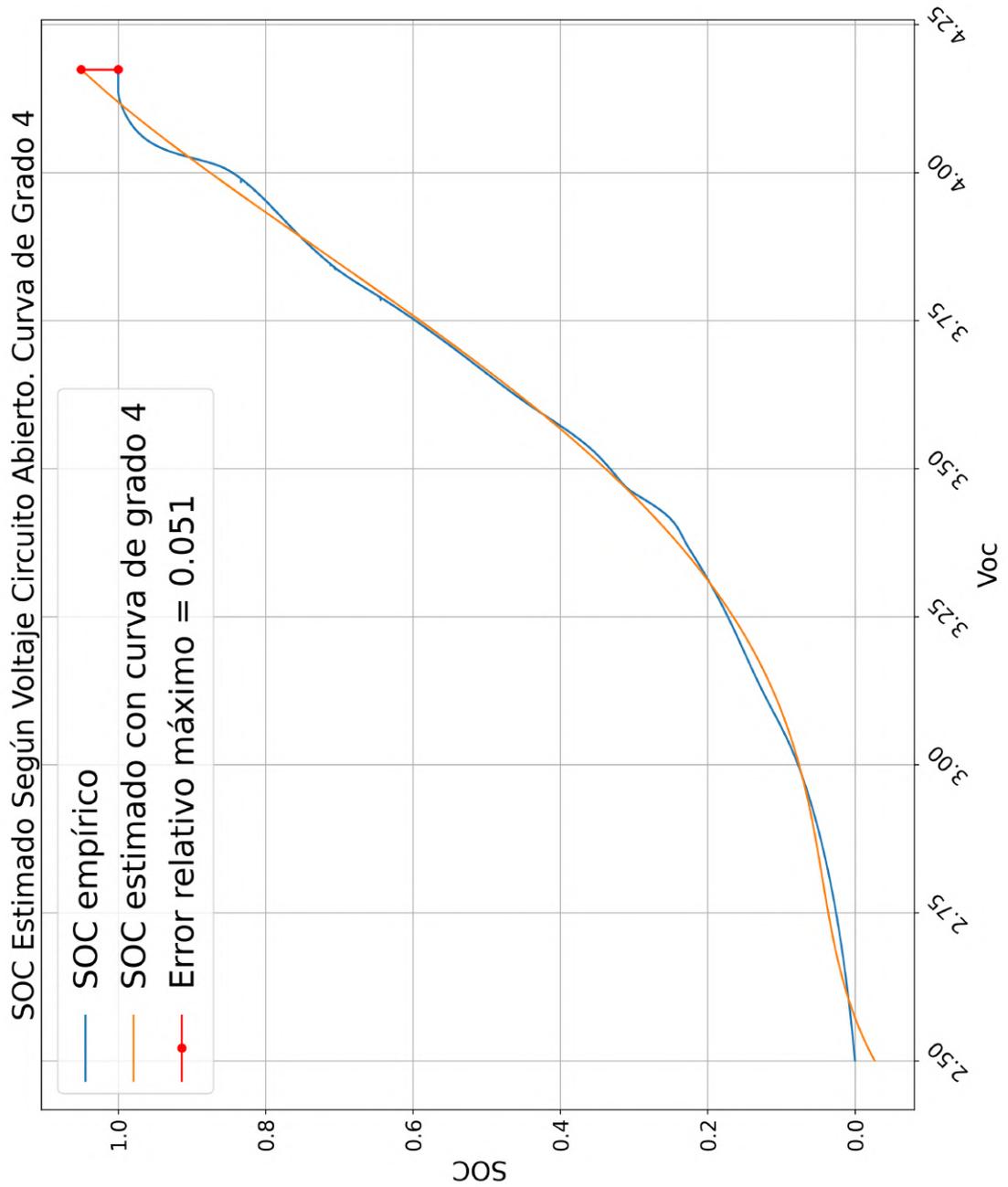


Figura B.5: Curva de regresión del SOC de grado 4.

## B.6. Regresión polinómica de la curva SOC. Grado 5

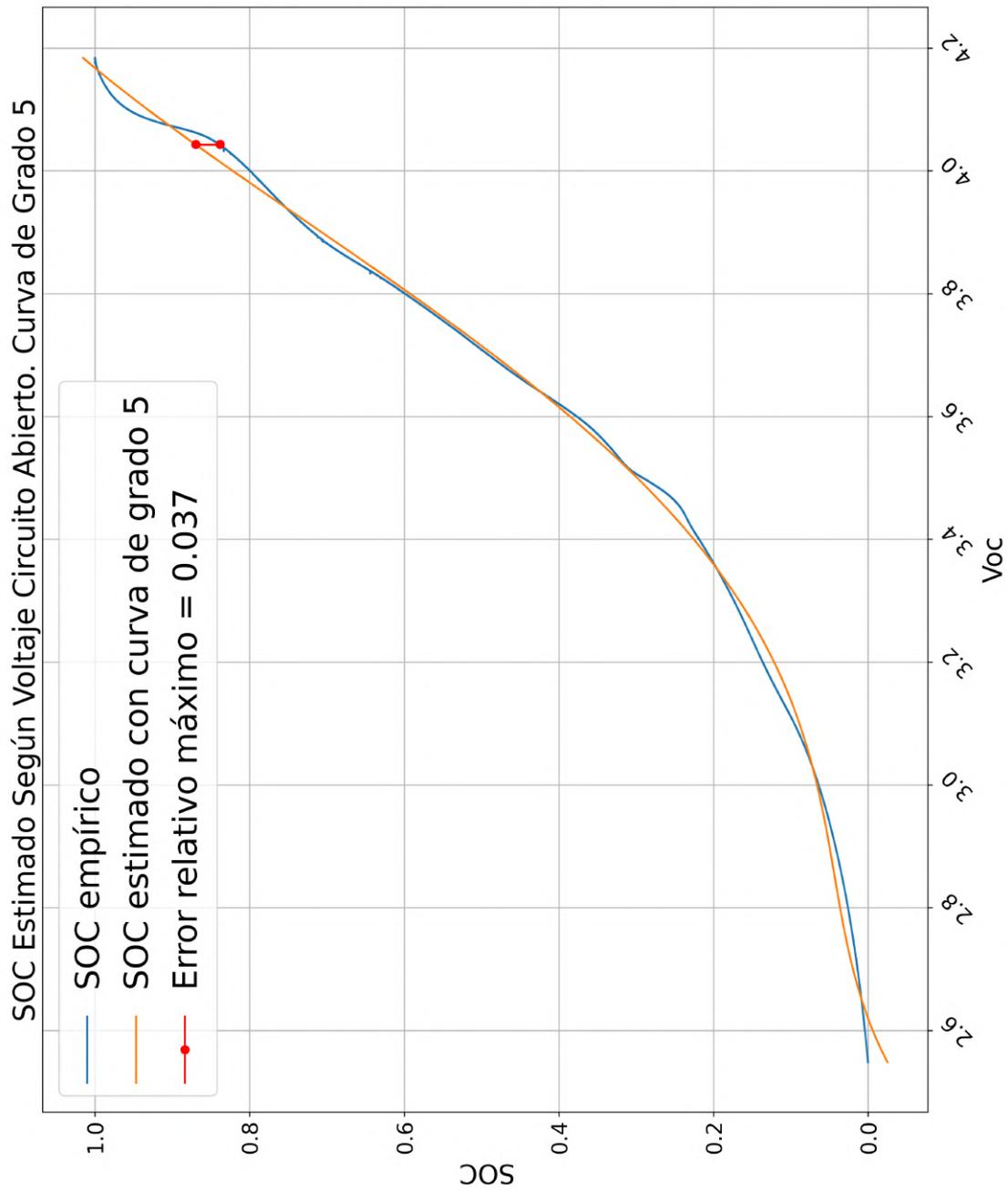


Figura B.6: Curva de regresión del SOC de grado 5.

## B.7. Regresión polinómica de la curva SOC. Grado 6

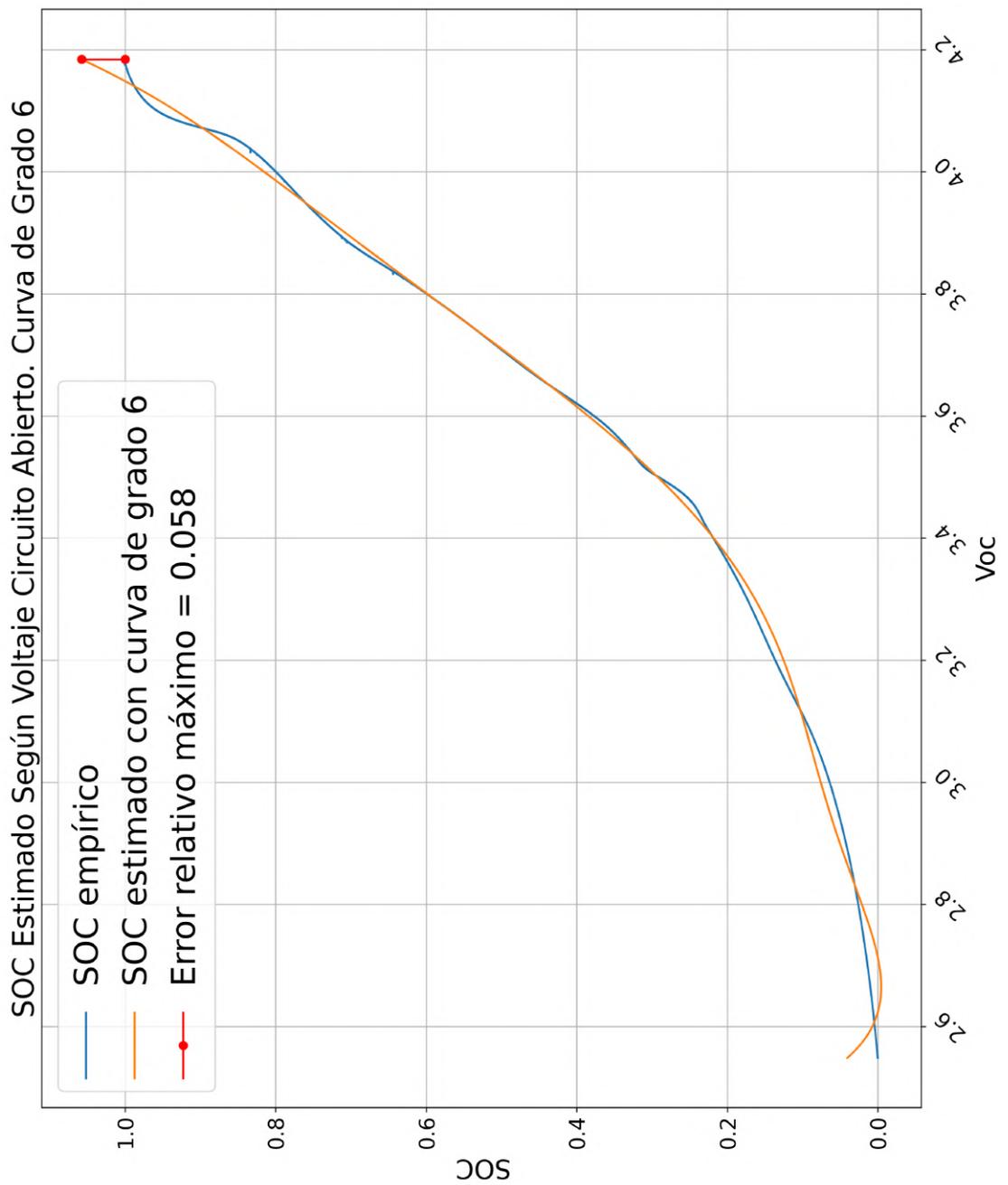


Figura B.7: Curva de regresión del SOC de grado 6.

# Apéndice C

## Gráficos de consumo del nodo

A continuación se presentan las gráficas de consumo de potencia del nodo, empleadas en el Capítulo 6 para calcular la potencia a suministrar.

### C.1. Consumo de corriente del día 21 de abril de 2021

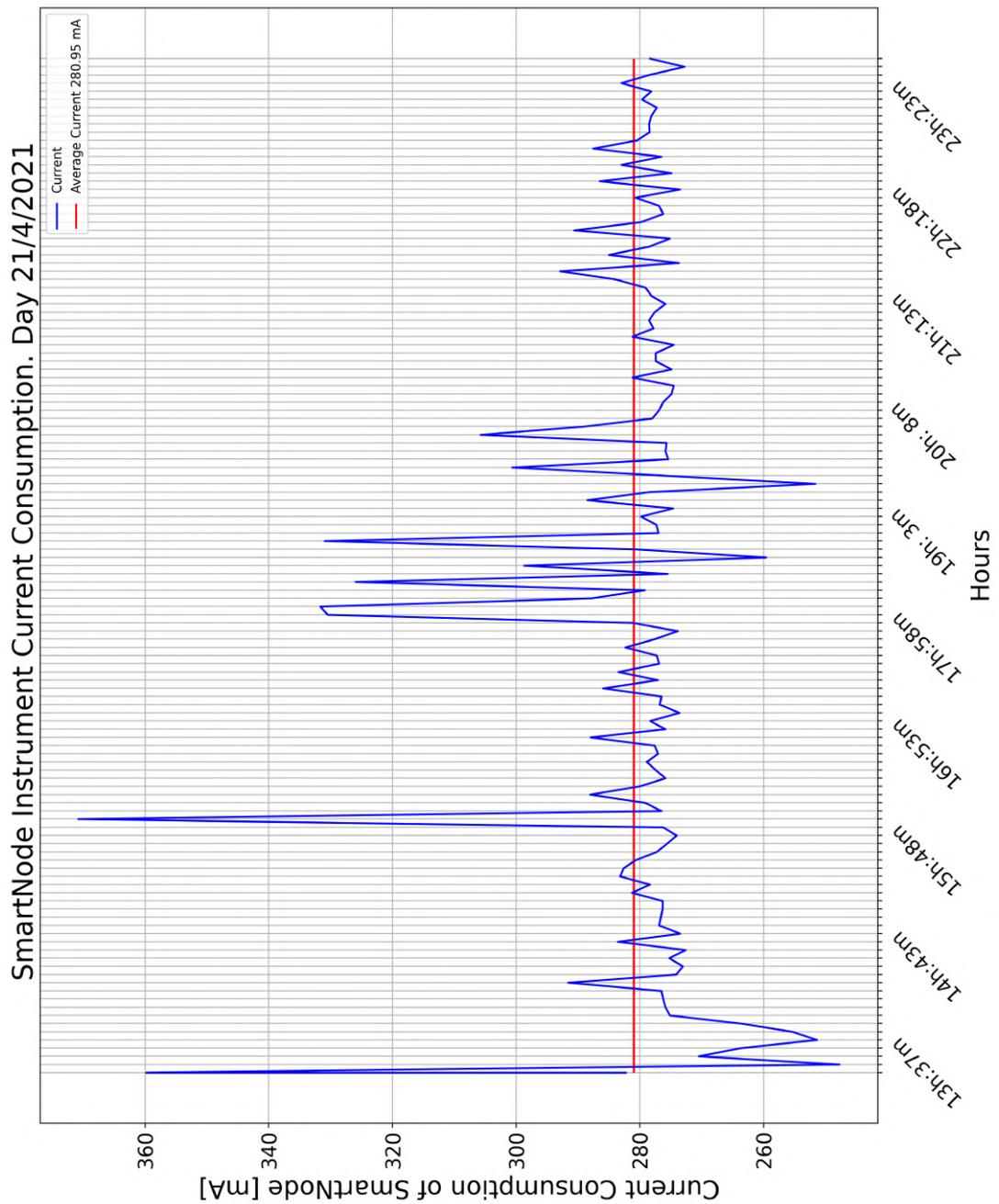


Figura C.1: Consumo de corriente del sistema *SmartNode* el 21/04/21.

## C.2. Consumo de potencia del día 21 de abril de 2021

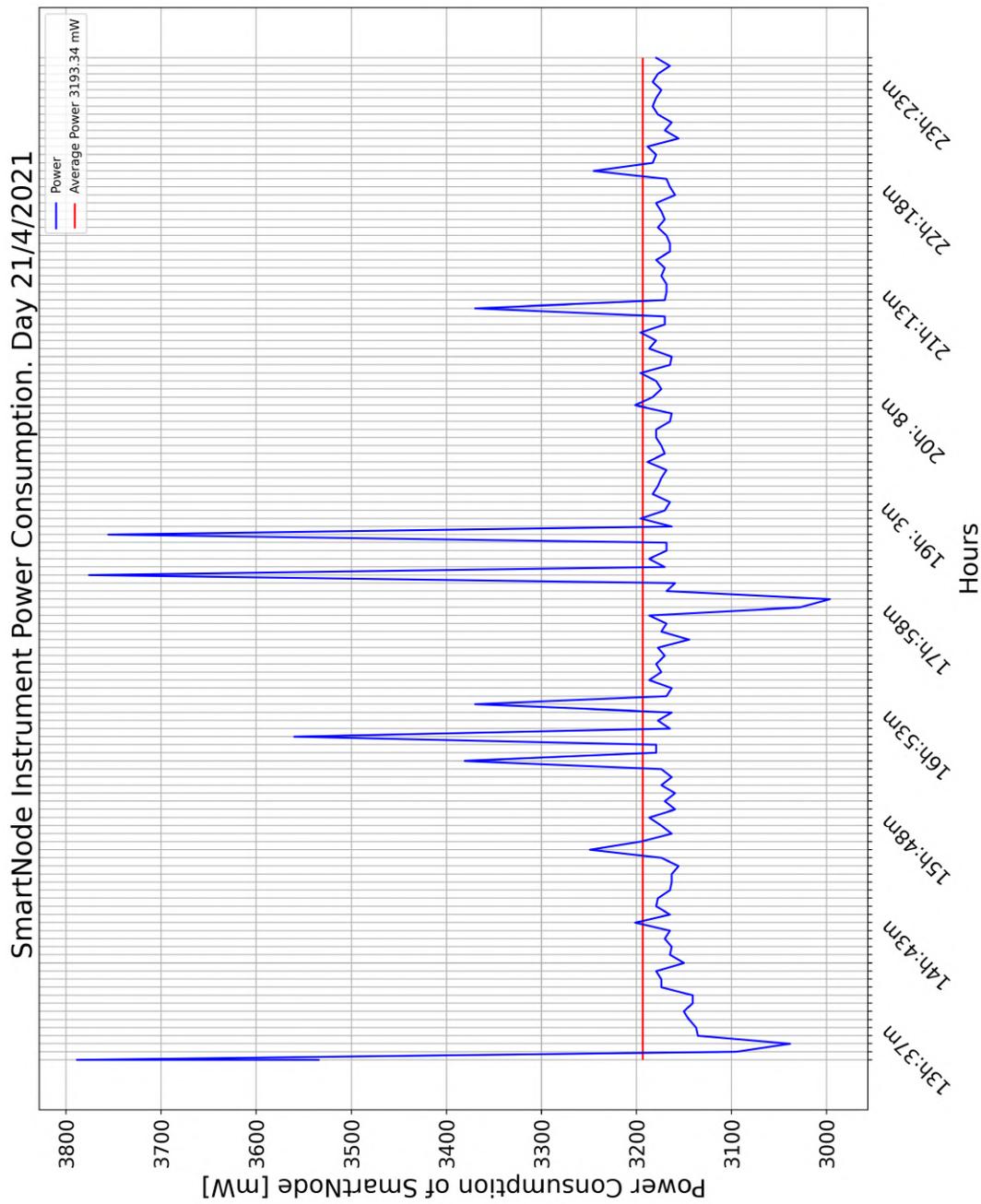


Figura C.2: Consumo de potencia del sistema *SmartNode* el 21/04/21.

### C.3. Voltaje del sistema el día 21 de abril de 2021

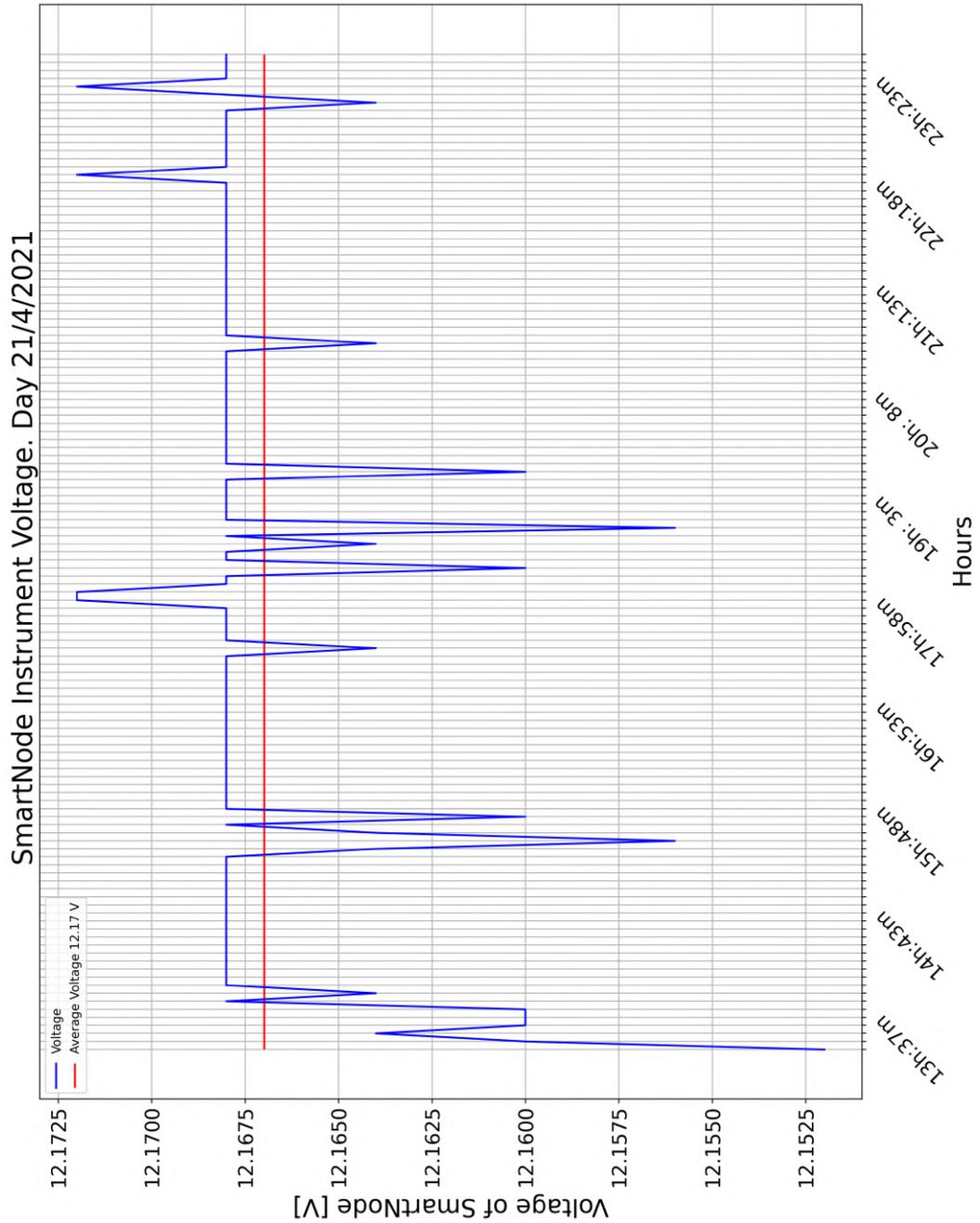


Figura C.3: Voltaje del sistema *SmartNode* el 21/04/21.

### C.4. Temperatura del sistema el día 21 de abril de 2021

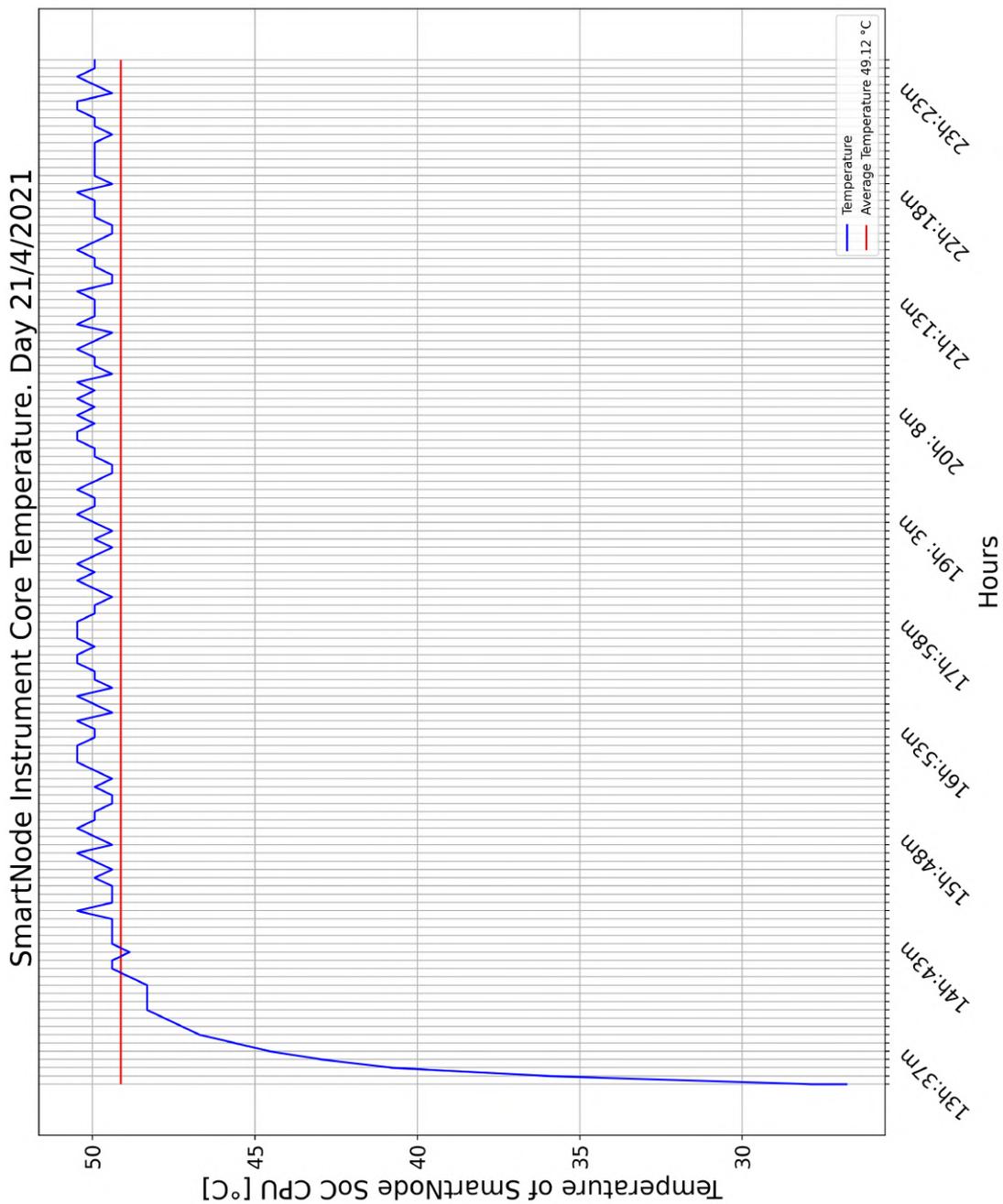


Figura C.4: Temperatura del sistema *SmartNode* el 21/04/21.

### C.5. Consumo de corriente del día 22 de abril de 2021

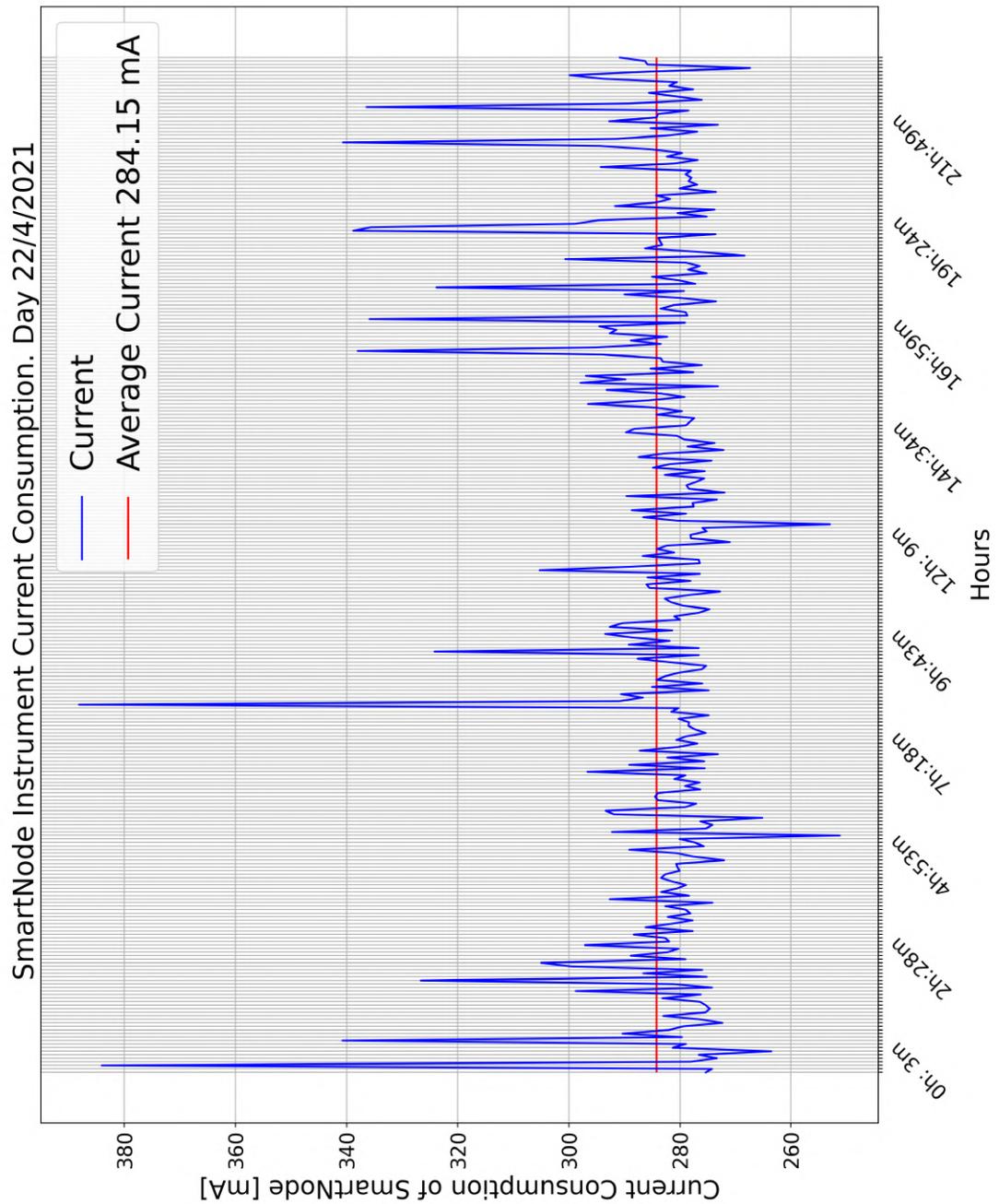


Figura C.5: Consumo de corriente del sistema *SmartNode* el 22/04/21.

### C.6. Consumo de potencia del día 22 de abril de 2021

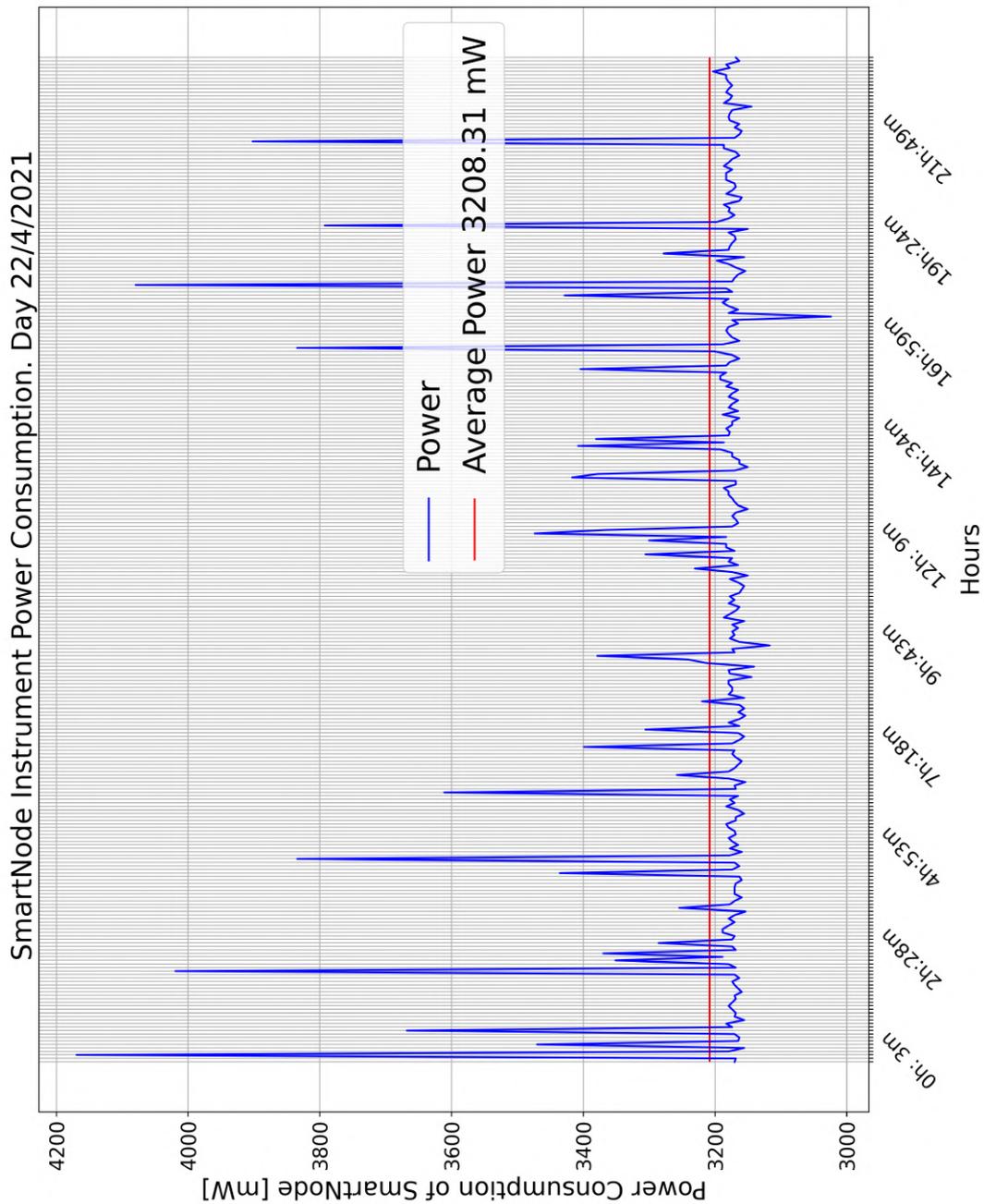


Figura C.6: Consumo de potencia del sistema *SmartNode* el 22/04/21.

### C.7. Voltaje del sistema el día 22 de abril de 2021

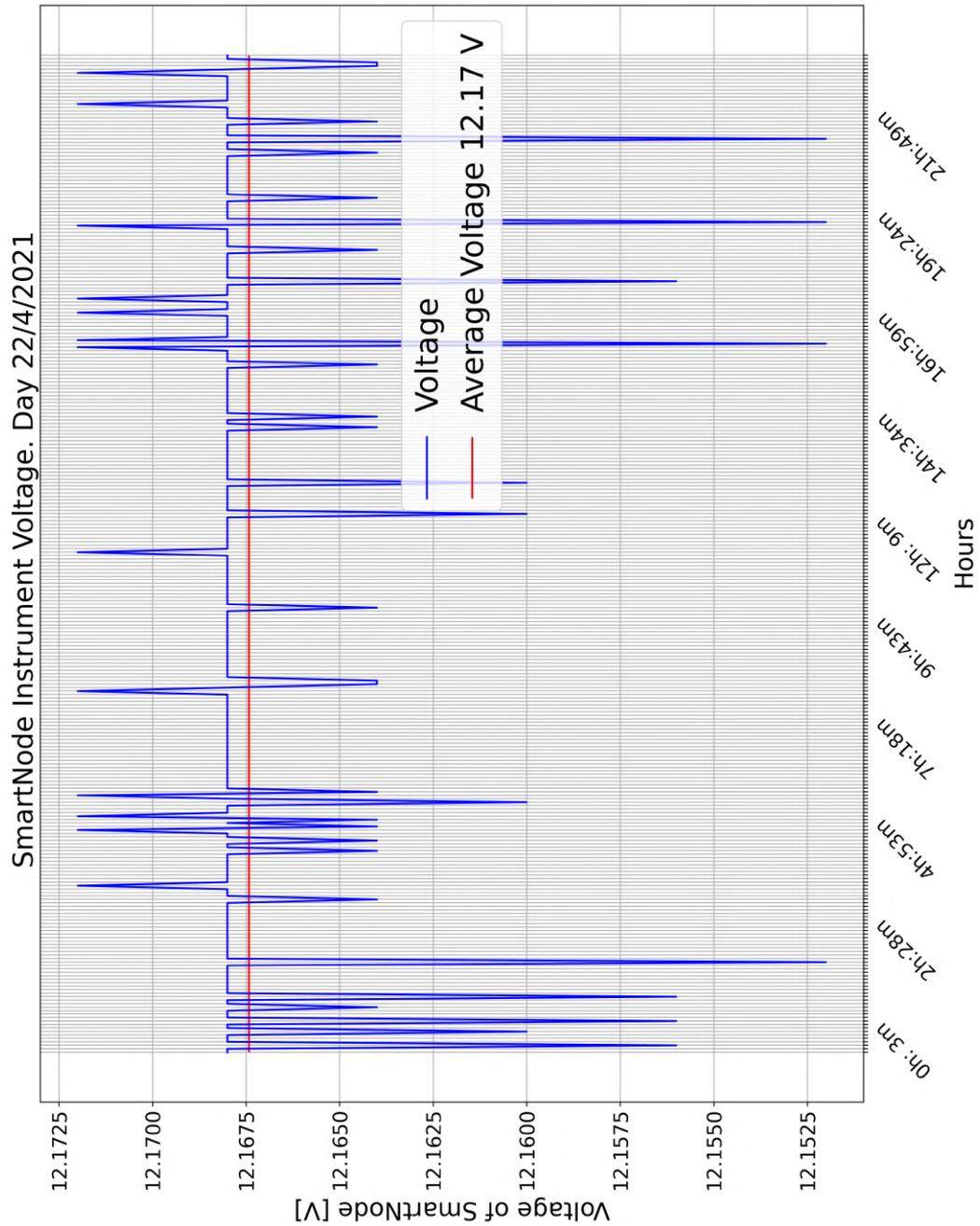


Figura C.7: Voltaje del sistema *SmartNode* el 22/04/21.

### C.8. Temperatura del sistema el día 22 de abril de 2021

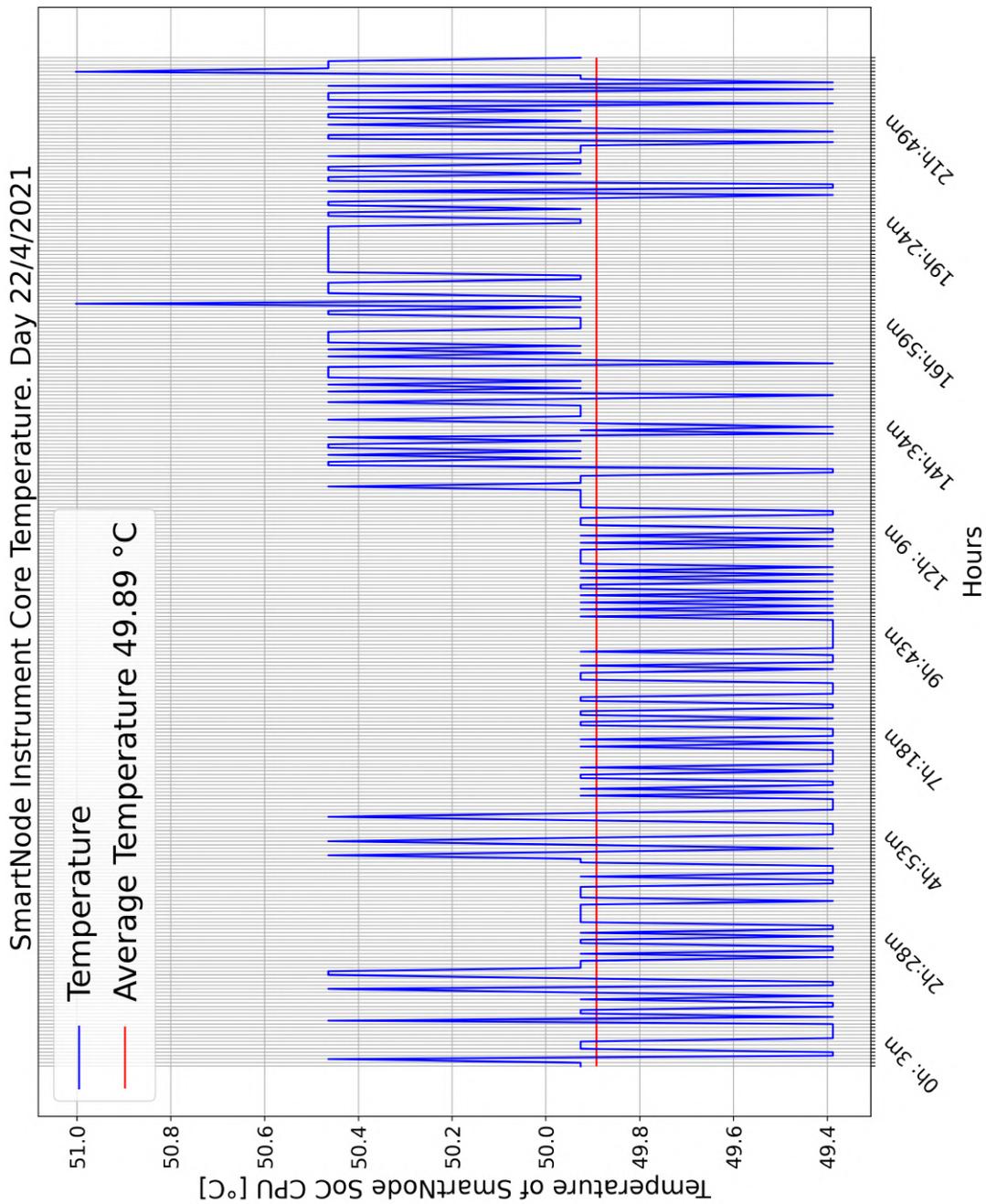


Figura C.8: Temperatura del sistema *SmartNode* el 22/04/21.



# Apéndice D

## Gráficos de *tests* del sistema

A continuación se presentan las gráficas de descarga y carga de la batería mostradas en el Capítulo 9. Se comienza mostrando los datos del *test* de descarga con  $44 \Omega$ , posteriormente con  $11 \Omega$ , la descarga a  $10 A$  y finalmente, la carga a  $15 A$ .

### D.1. Corriente extraída de la batería. *Test* de 44 $\Omega$

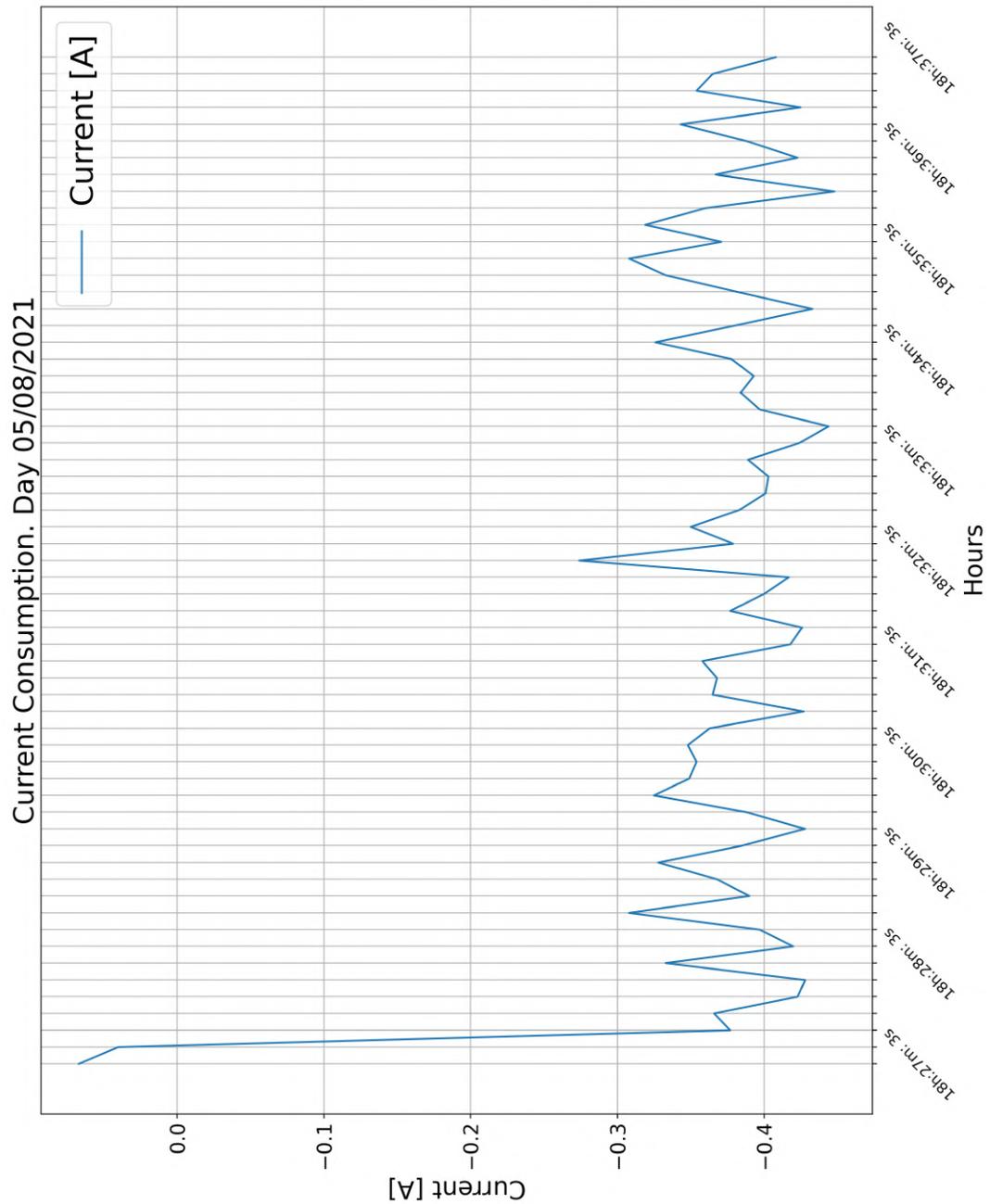


Figura D.1: Consumo de corriente de la batería con 44  $\Omega$ .

## D.2. Tensión de las celdas. Test de 44 Ω

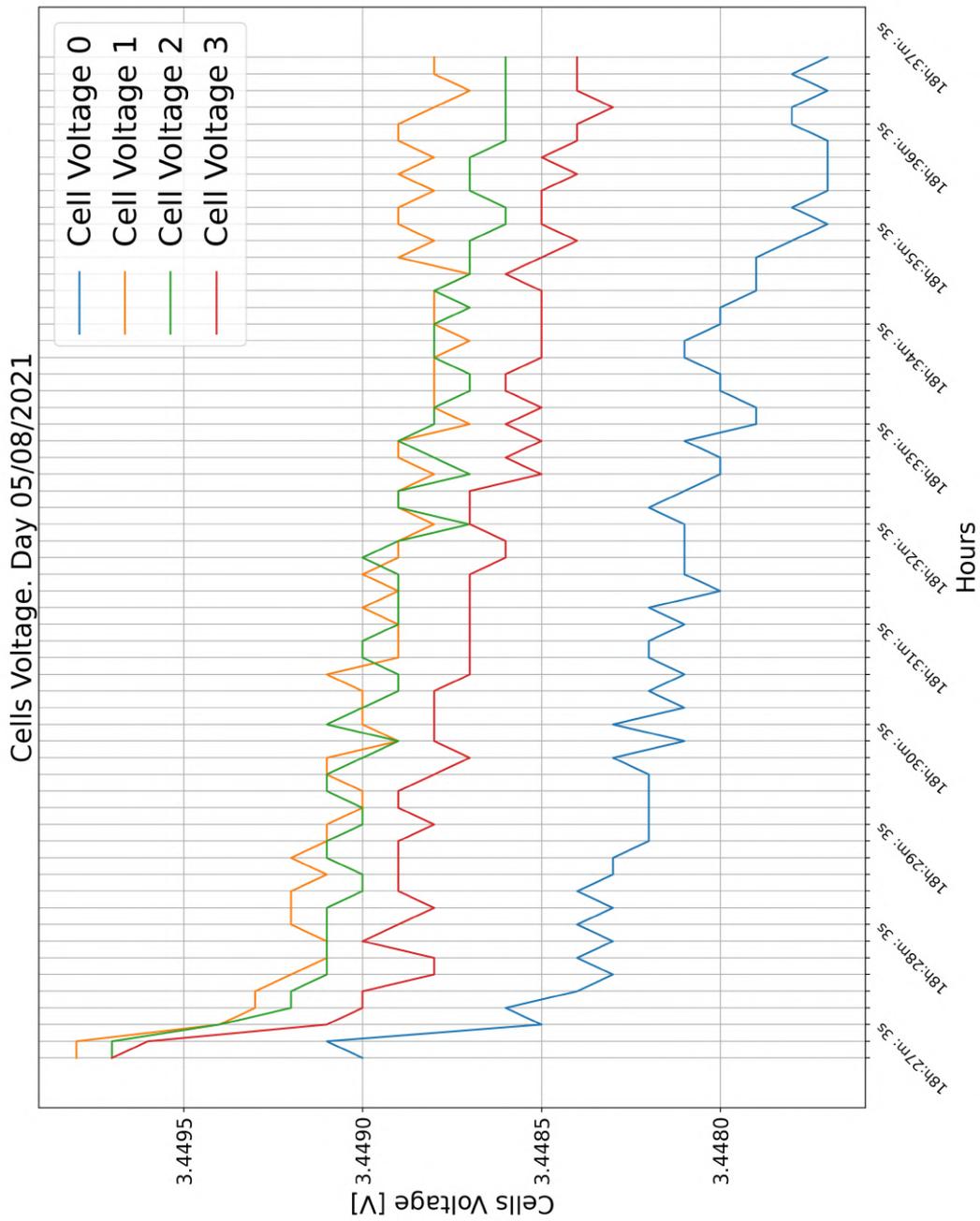


Figura D.2: Tensión de las celdas con *test* de 44 Ω.

### D.3. Diferencia de tensión de las celdas. *Test* de 44 $\Omega$

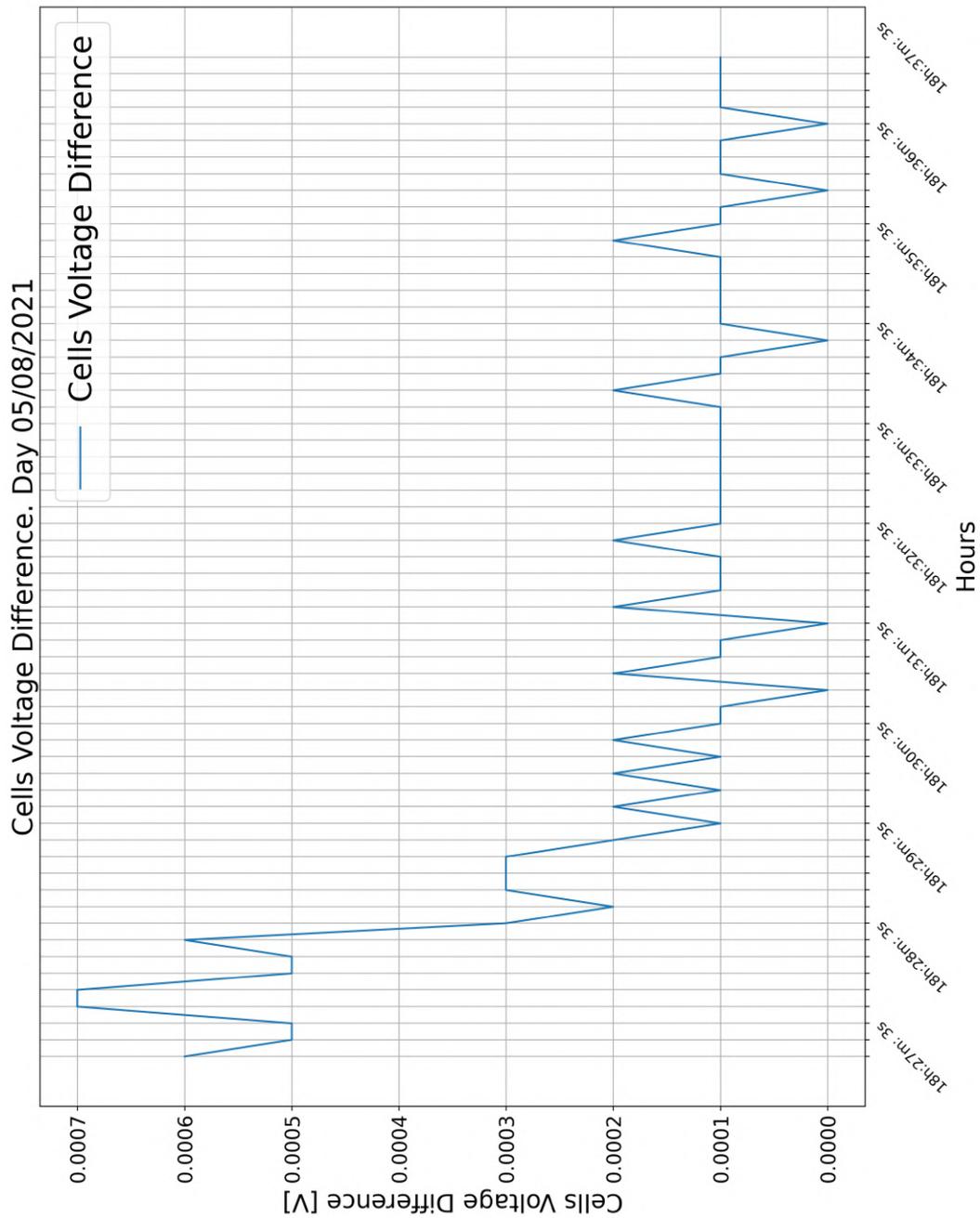


Figura D.3: Diferencia de tensión de las celdas con *test* de 44  $\Omega$ .

### D.4. SOC de la batería. *Test* de 44 Ω

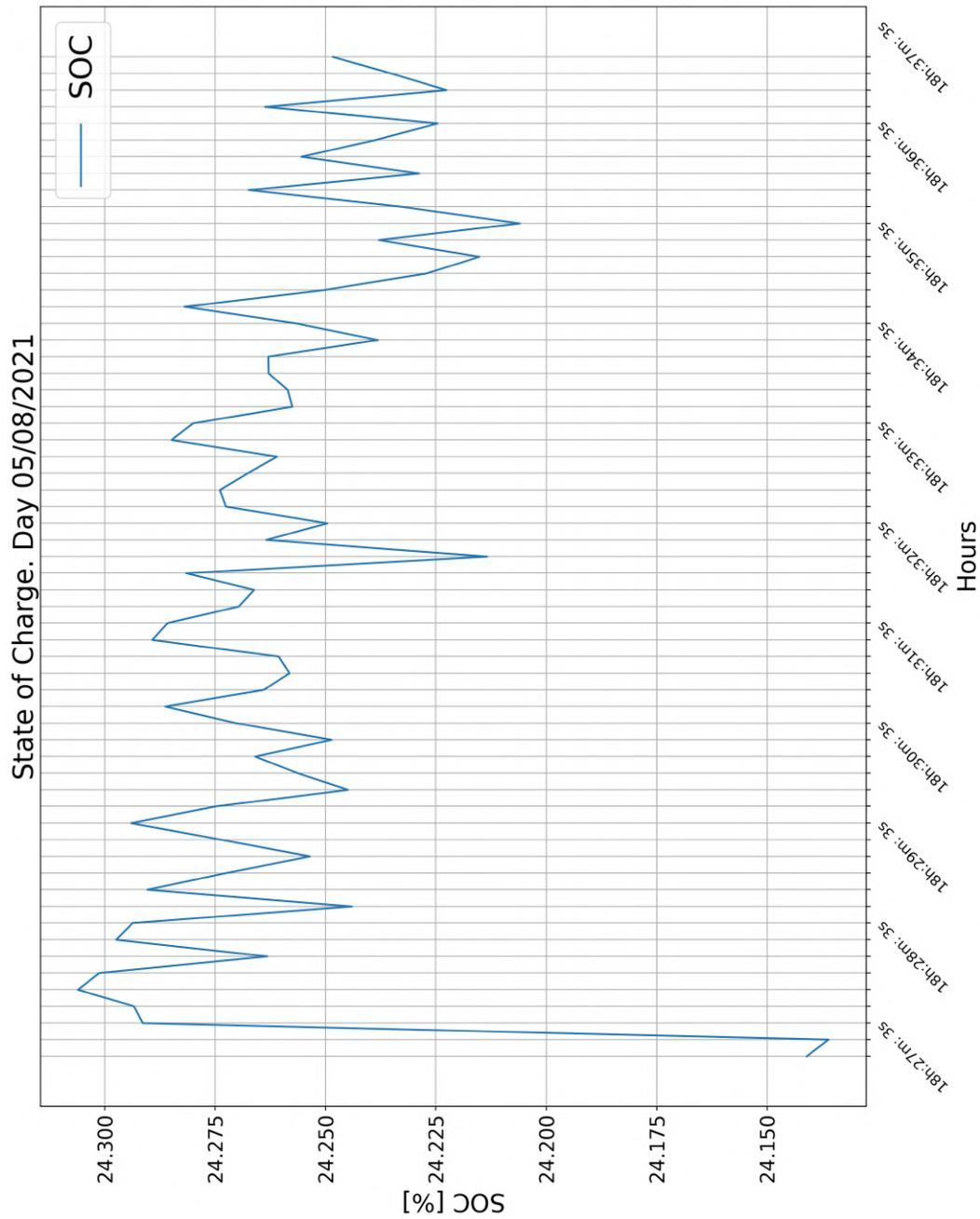


Figura D.4: SOC de la batería con *test* de 44 Ω.

### D.5. Temperatura de la batería. *Test* de 44 $\Omega$

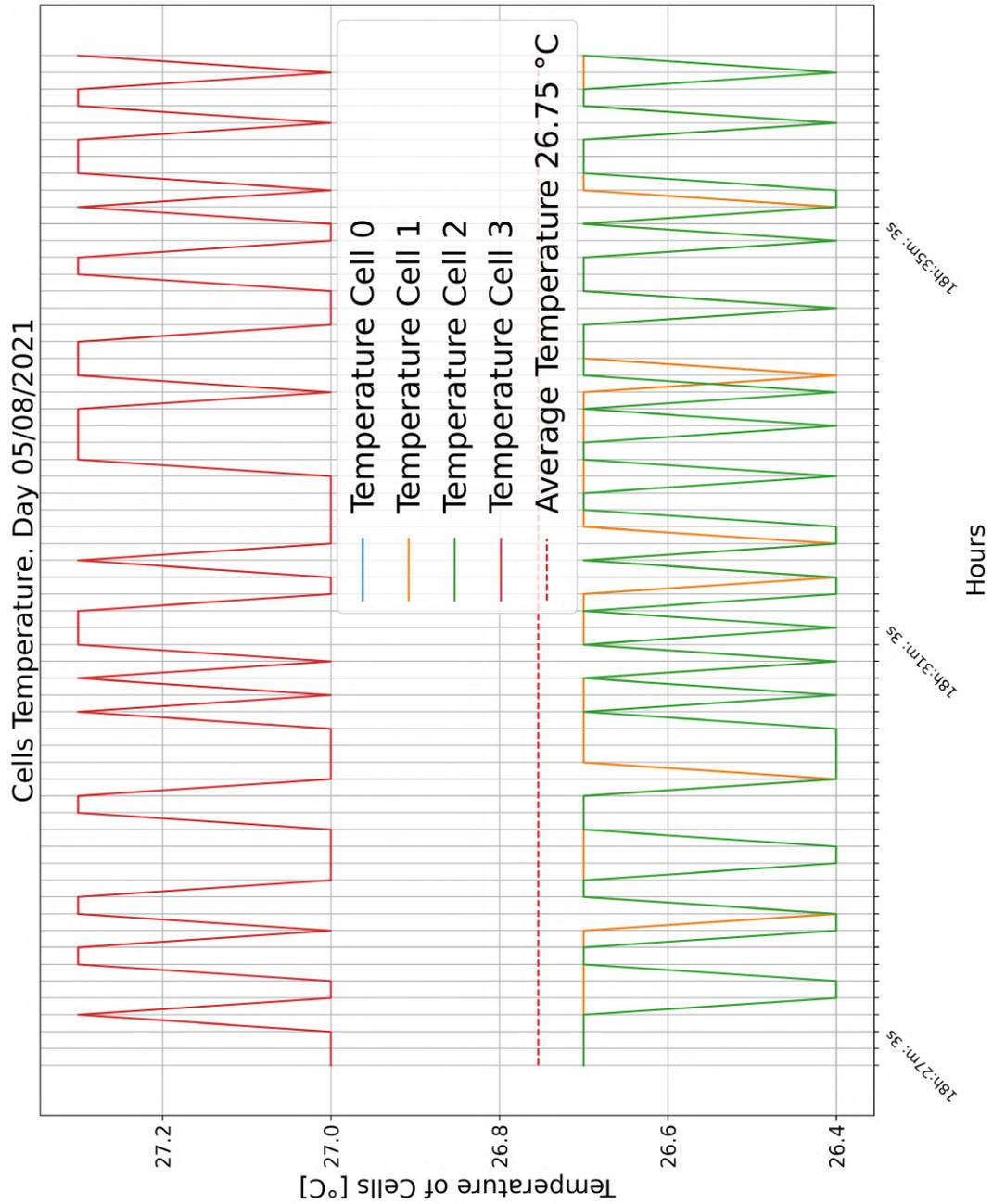


Figura D.5: Temperatura de la batería con *test* de 44  $\Omega$ .

### D.6. Corriente extraída de la batería. *Test* de 11 $\Omega$

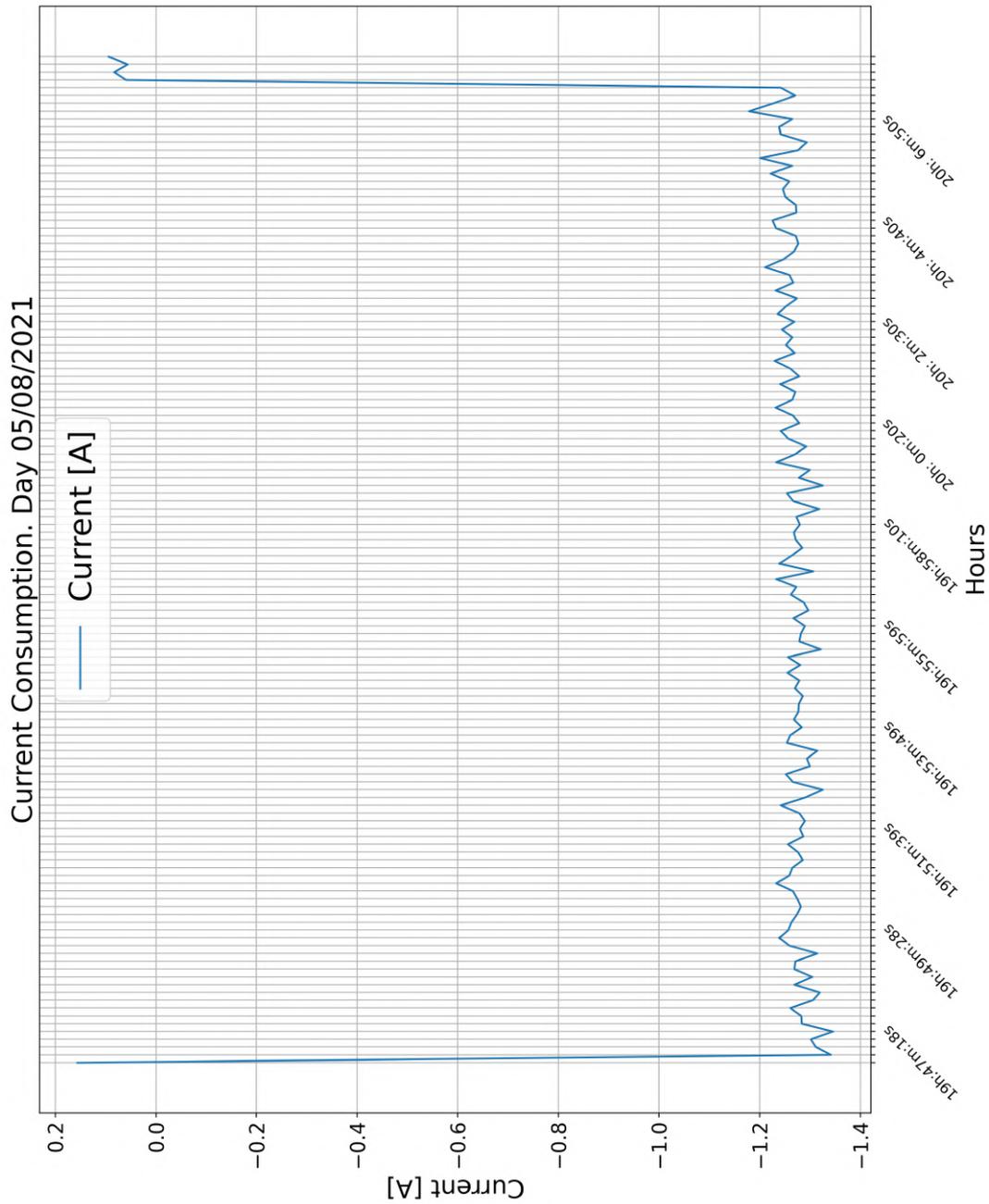


Figura D.6: Consumo de corriente de la batería con 11  $\Omega$ .

## D.7. Tensión de las celdas. *Test* de 11 $\Omega$

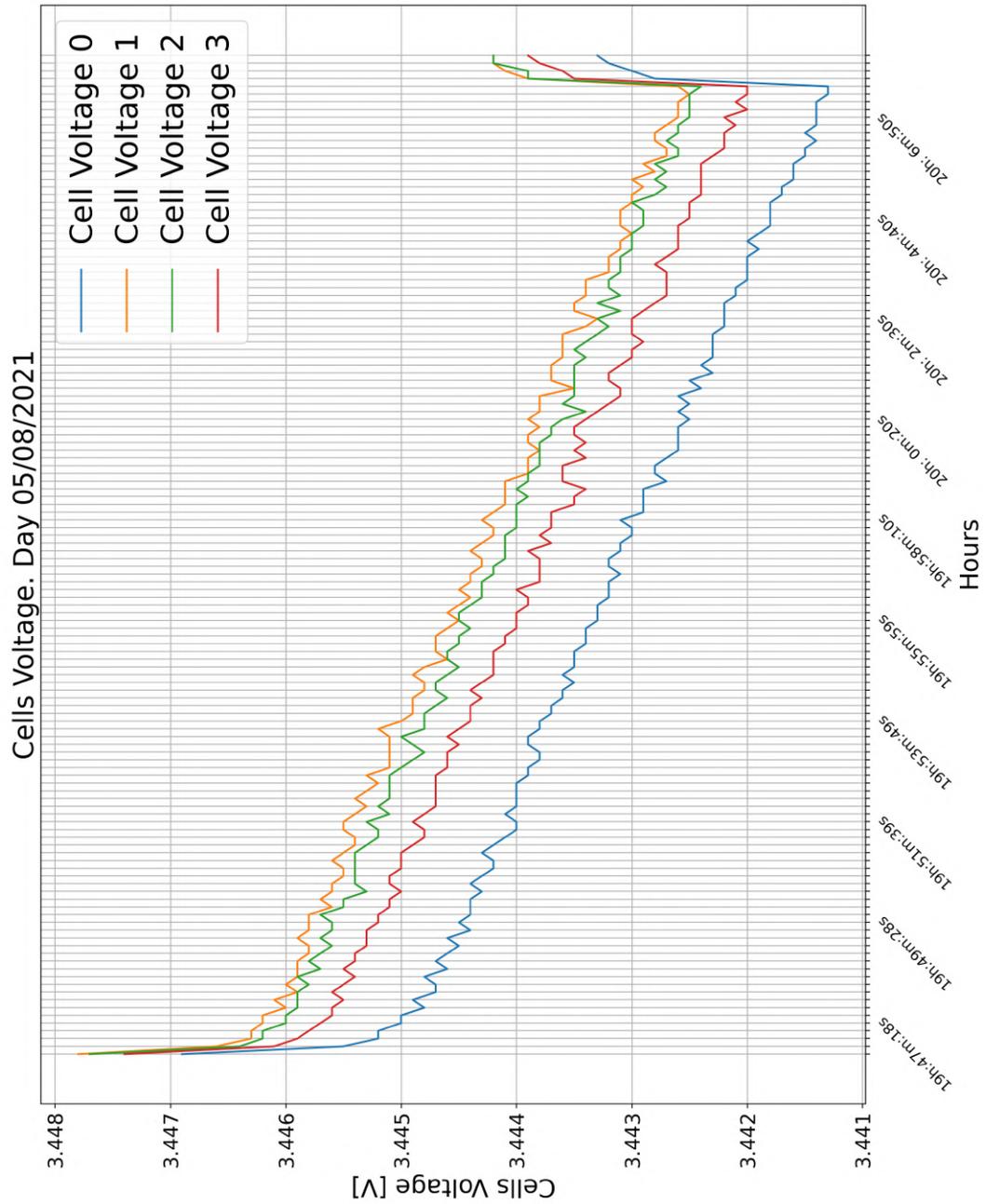


Figura D.7: Tensión de las celdas con *test* de 11  $\Omega$ .

### D.8. Diferencia de tensión de las celdas. Test de 11 $\Omega$

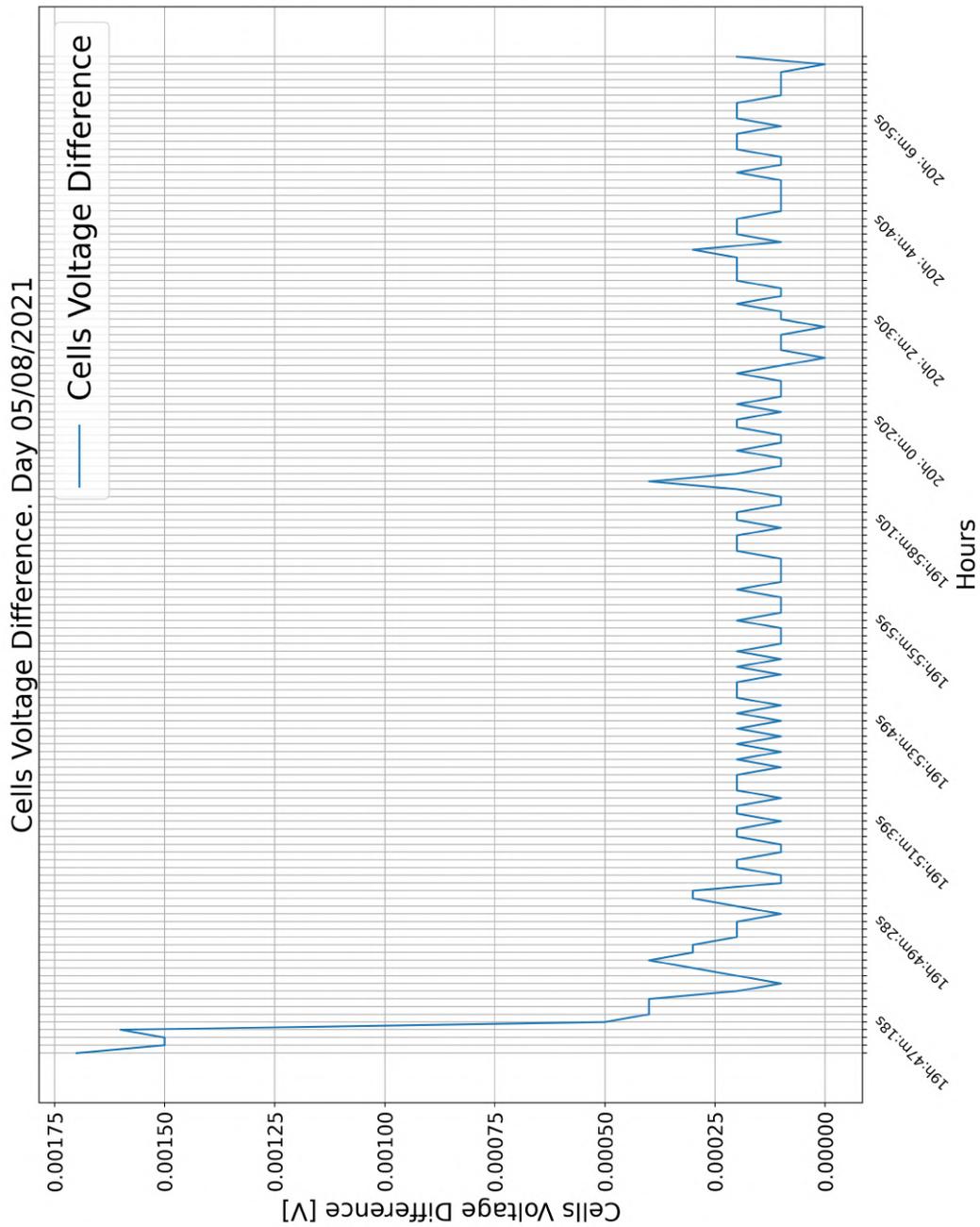
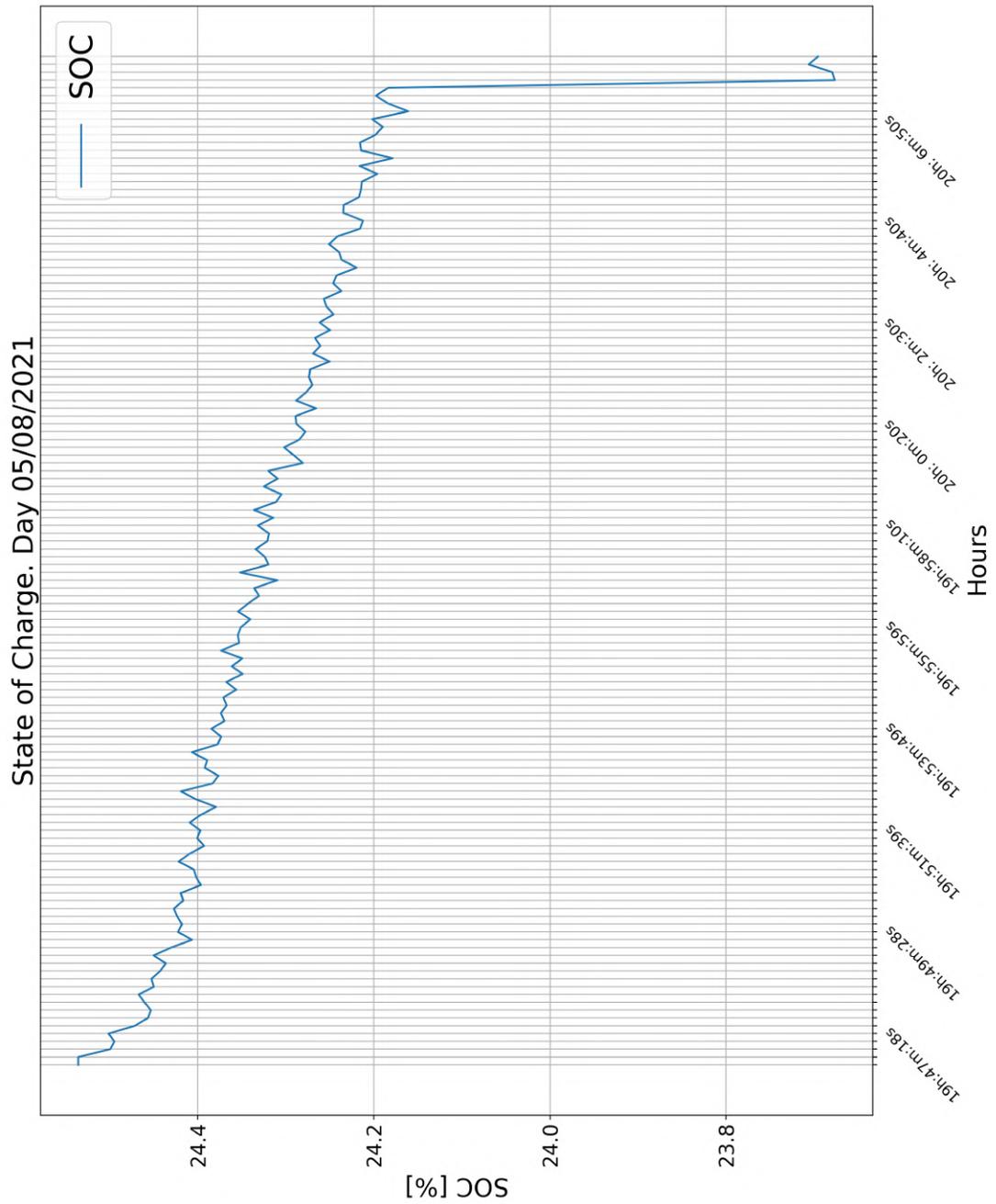


Figura D.8: Diferencia de tensión de las celdas con test de 11  $\Omega$ .

**D.9. SOC de la batería. *Test* de 11  $\Omega$** Figura D.9: SOC de la batería con *test* de 11  $\Omega$ .

### D.10. Temperatura de la batería. Test de 11 Ω

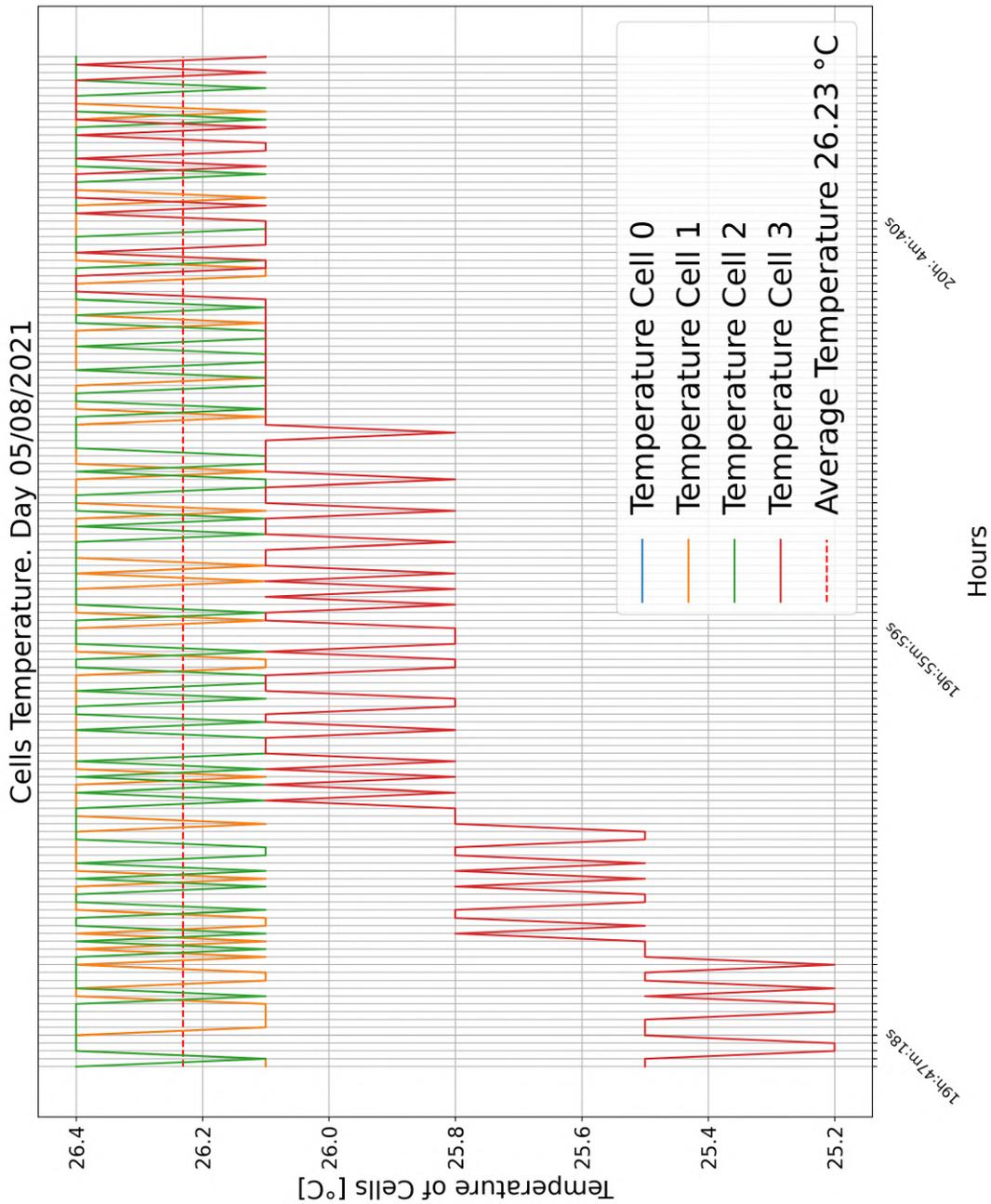


Figura D.10: Temperatura de la batería con *test* de 11 Ω.

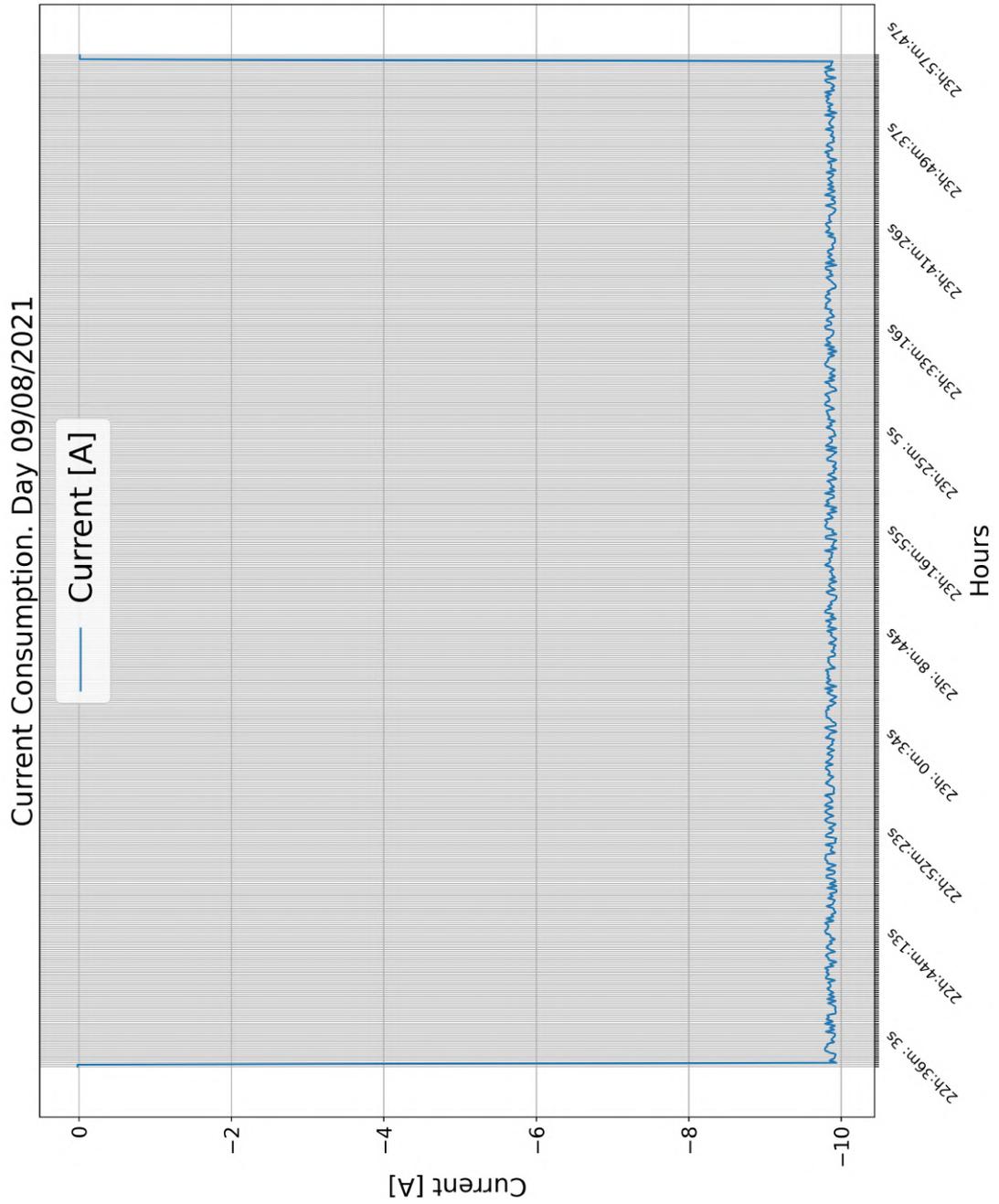
**D.11. Corriente extraída de la batería. Test de 10 A**

Figura D.11: Consumo de corriente de la batería con 10 A.

### D.12. Tensión de las celdas. Test de 10 A

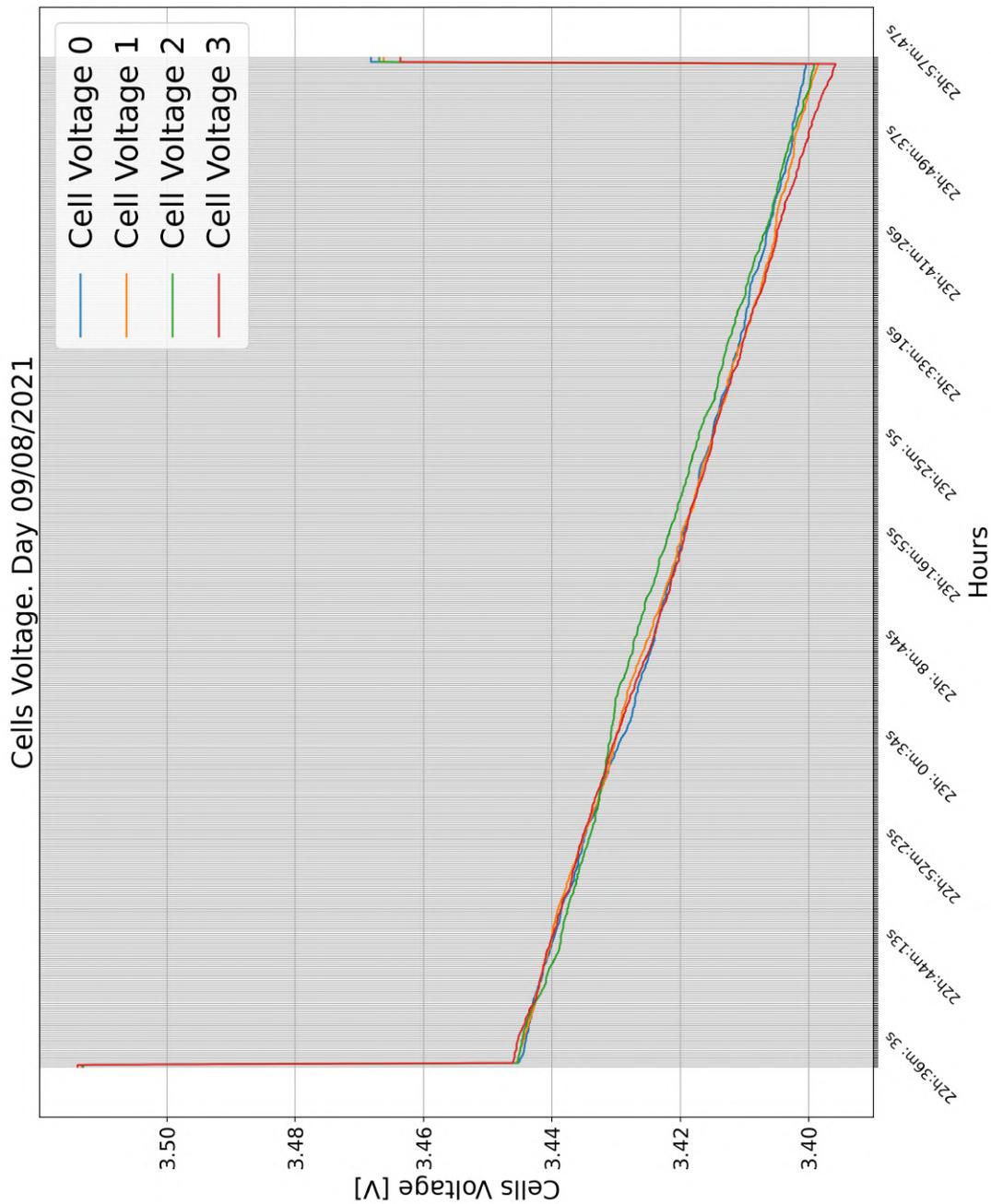


Figura D.12: Tensión de las celdas con test de 10 A.

### D.13. Diferencia de tensión de las celdas. *Test* de 10 A

#### A

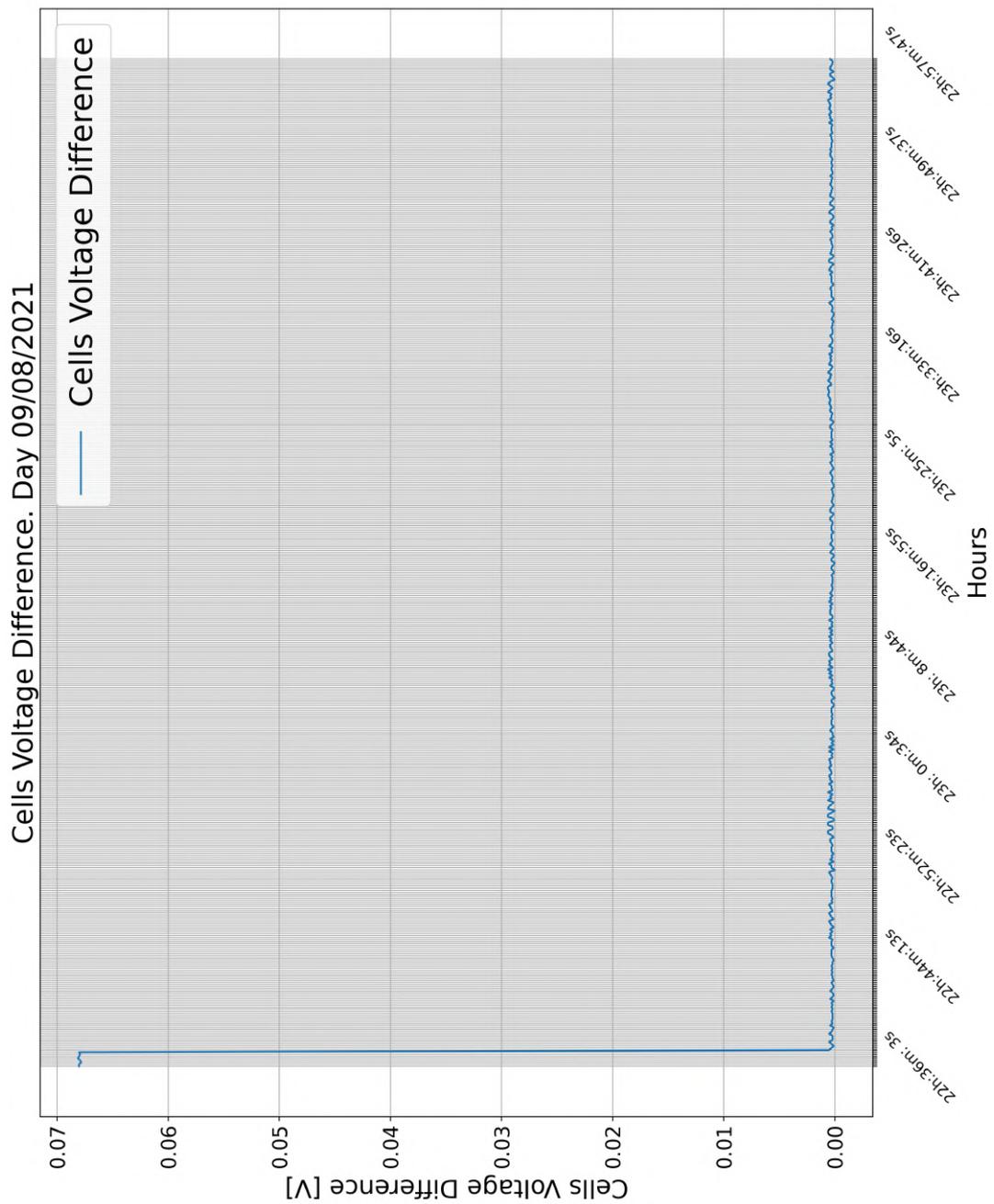


Figura D.13: Diferencia de tensión de las celdas con *test* de 10 A.

### D.14. SOC de la batería. Test de 10 A

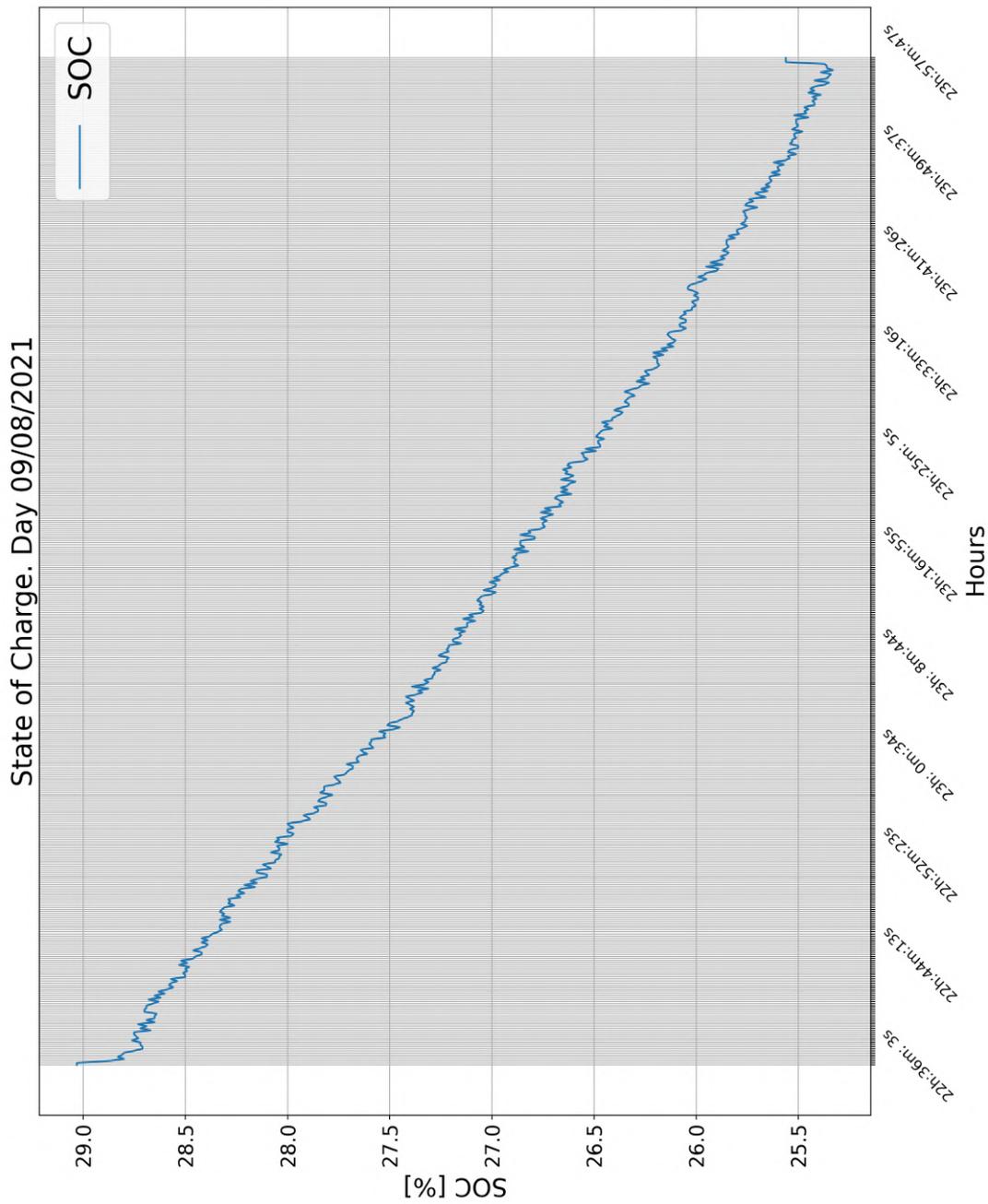


Figura D.14: SOC de la batería con *test* de 10 A.

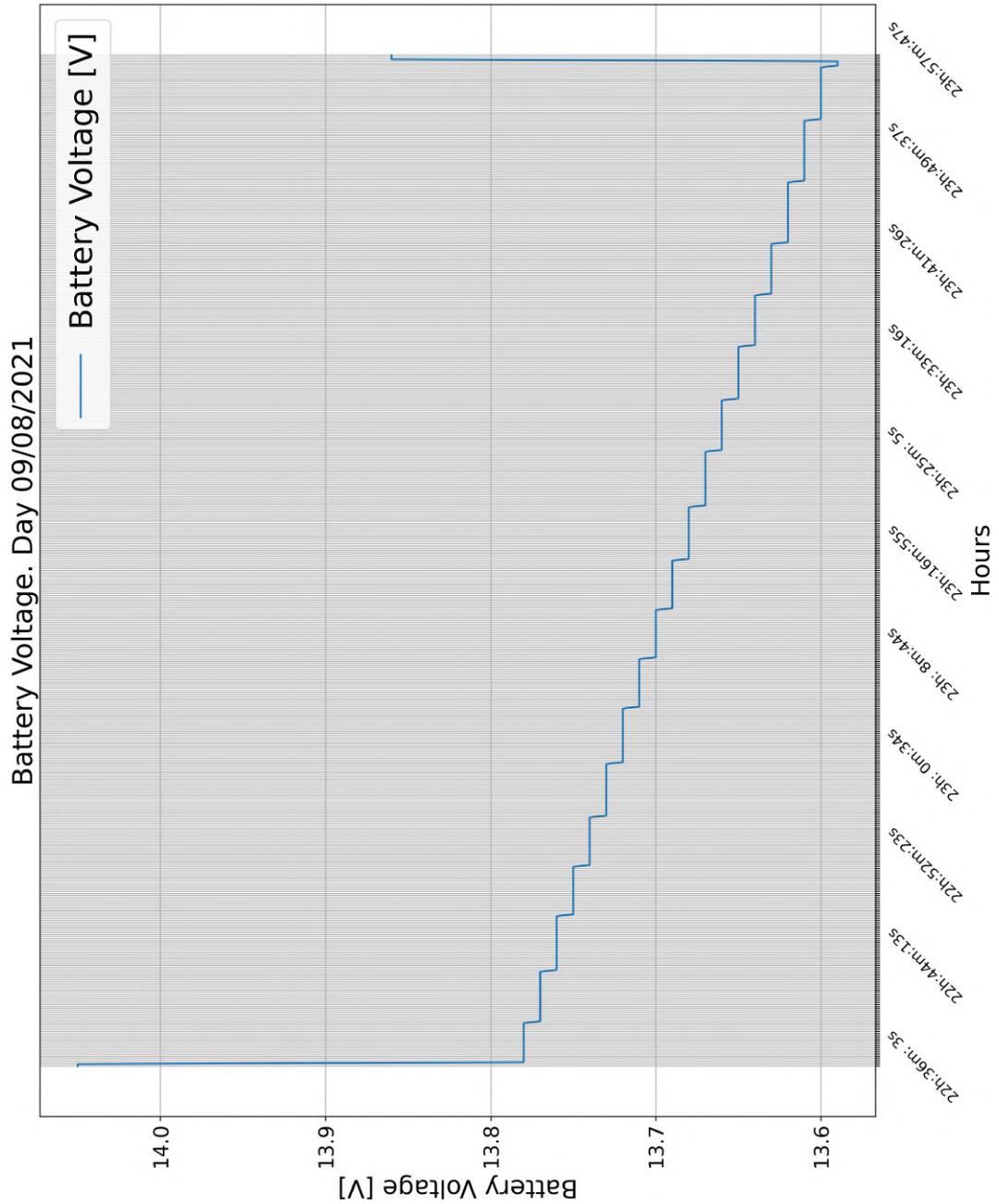
**D.15. Tensión de la batería. Test de 10 A**

Figura D.15: Tensión de la batería con test de 10 A.

### D.16. Temperatura de la batería. Test de 10 A

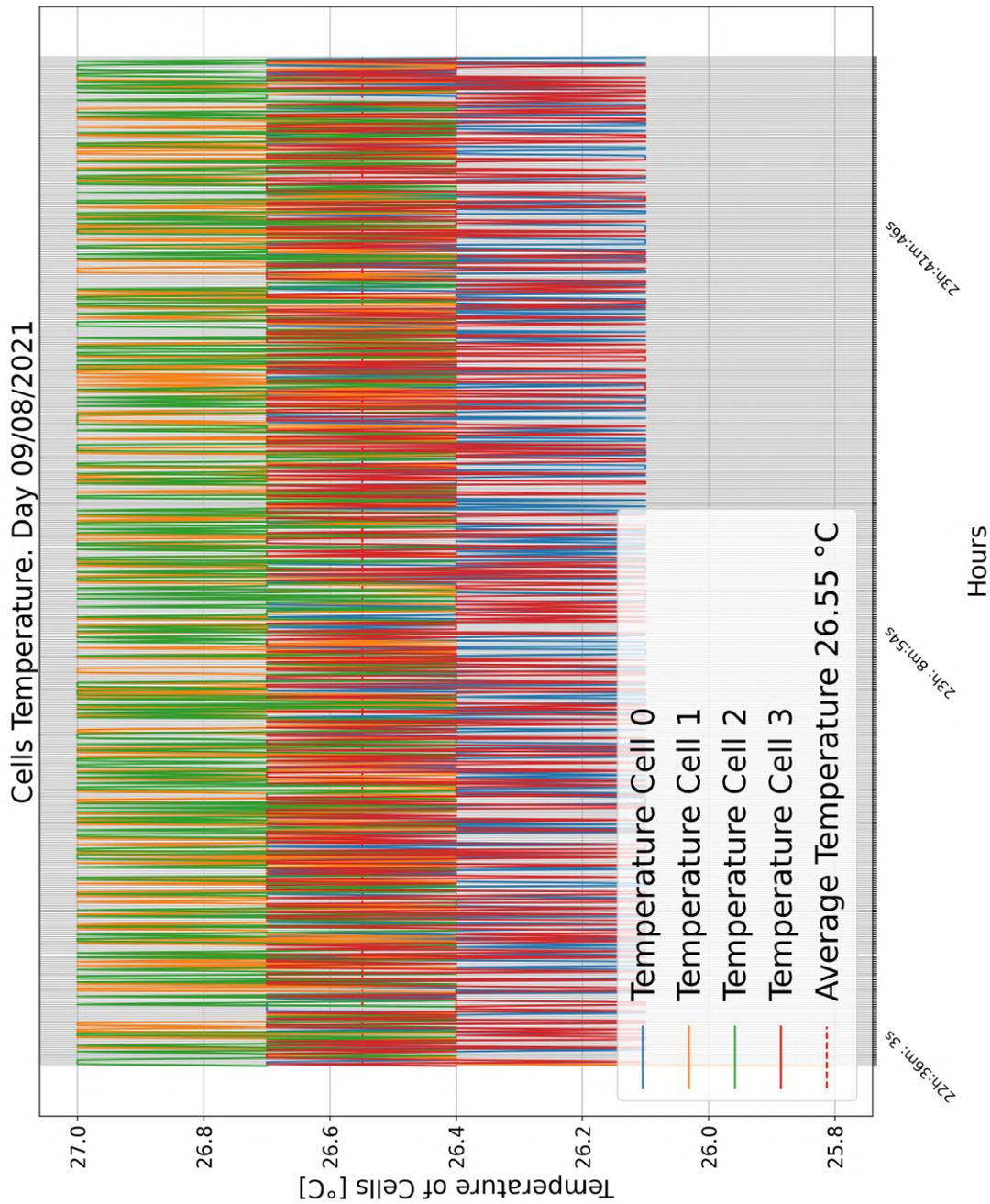


Figura D.16: Temperatura de la batería con *test* de 10 A.

### D.17. Corriente de carga a la batería. *Test de carga a 15 A*

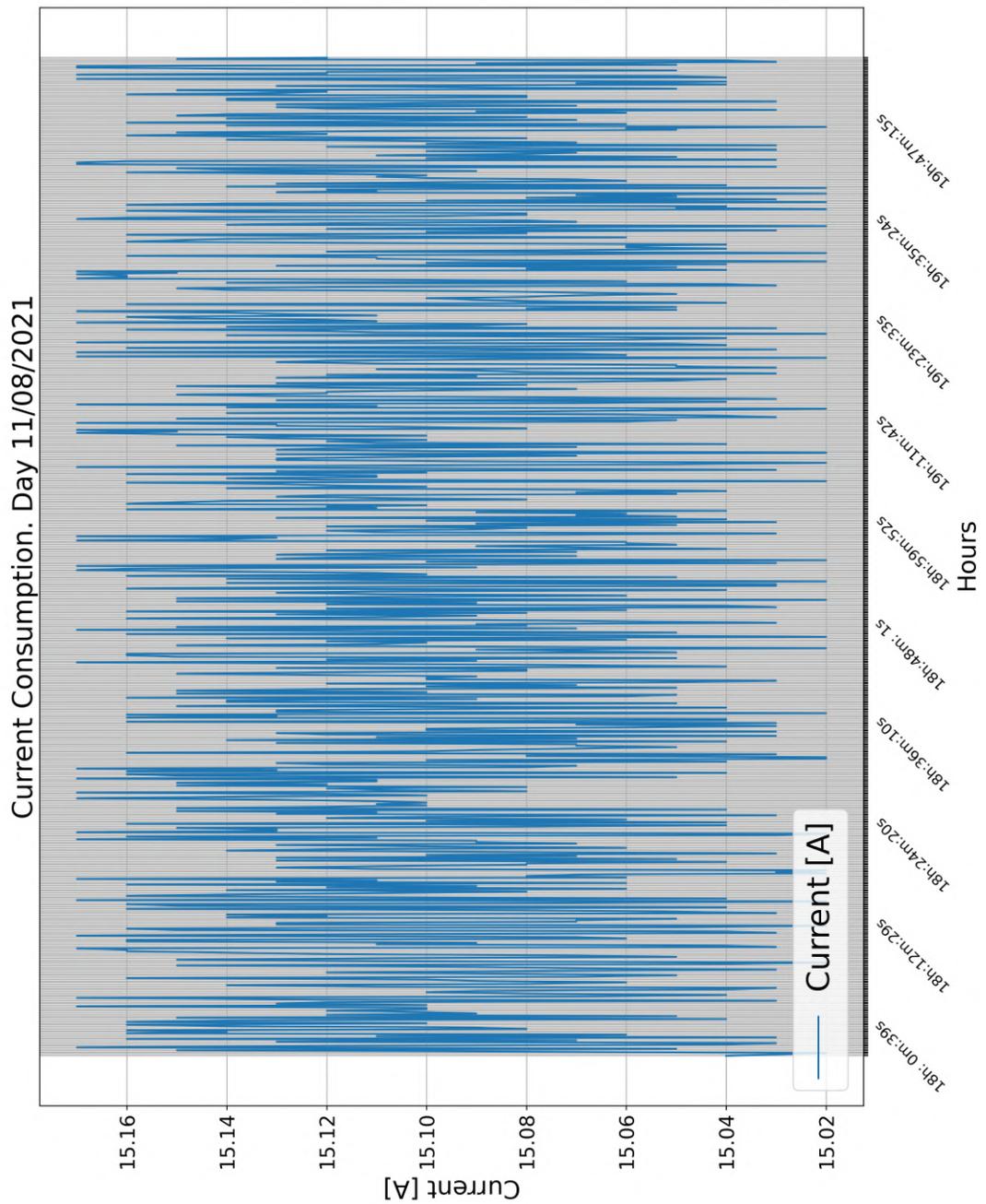


Figura D.17: Corriente de carga de la batería con 15 A.

### D.18. Tensión de las celdas. Test de carga a 15 A

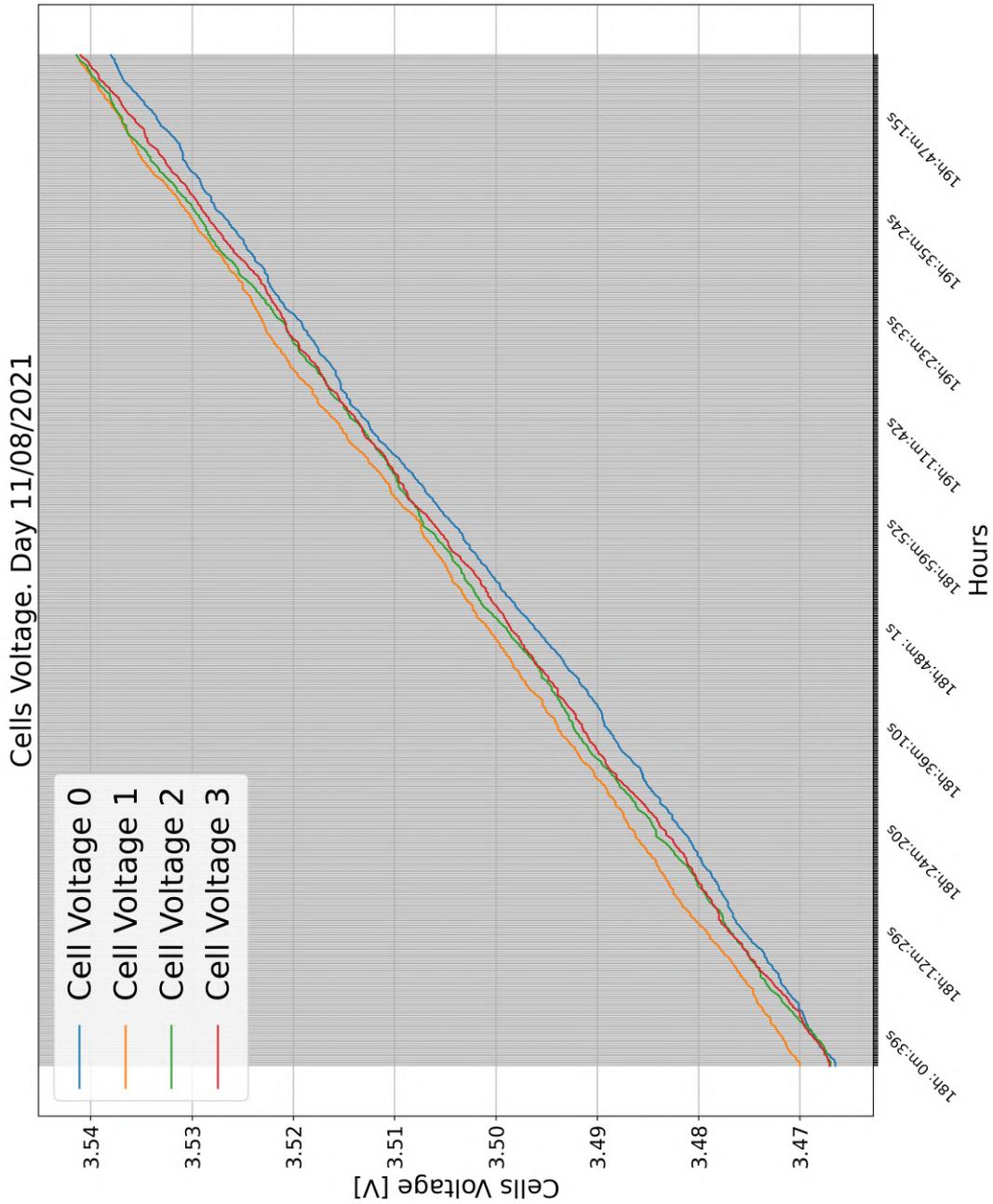


Figura D.18: Tensión de las celdas con test de 15 A.

## D.19. Diferencia de tensión de las celdas. *Test* de carga 15 A

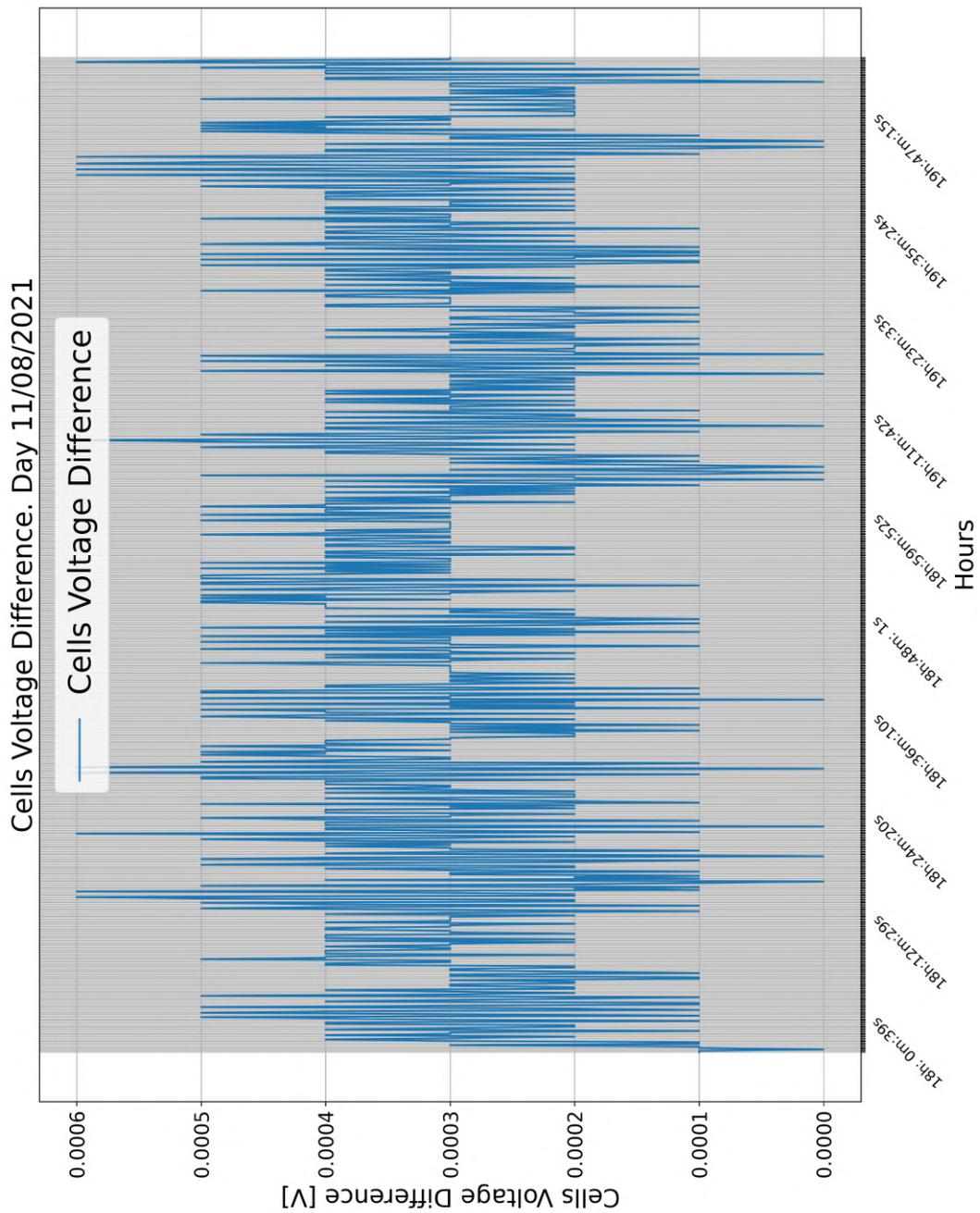


Figura D.19: Diferencia de tensión de las celdas con *test* de 15 A.

### D.20. SOC de la batería. *Test* de carga a 15 A

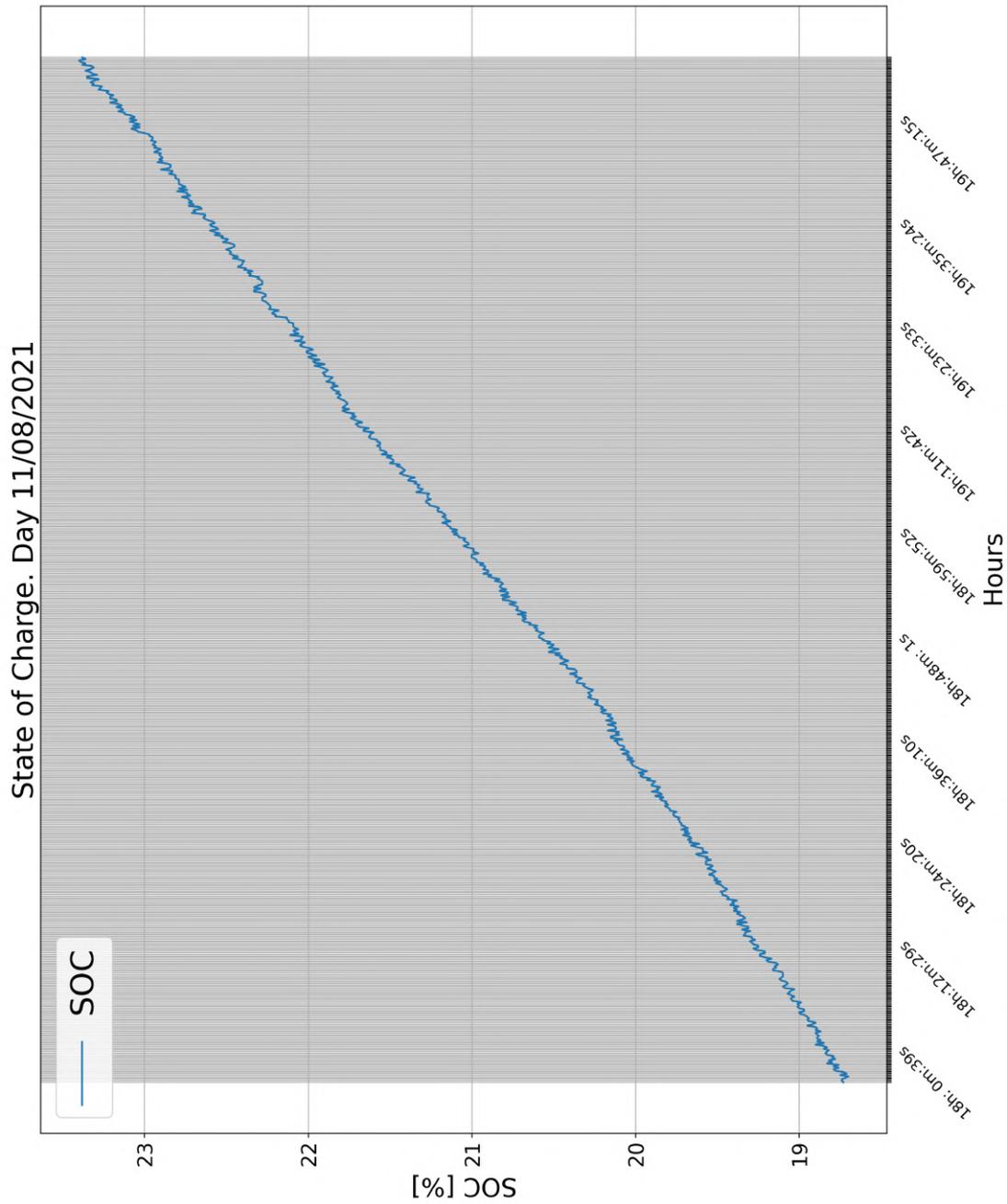
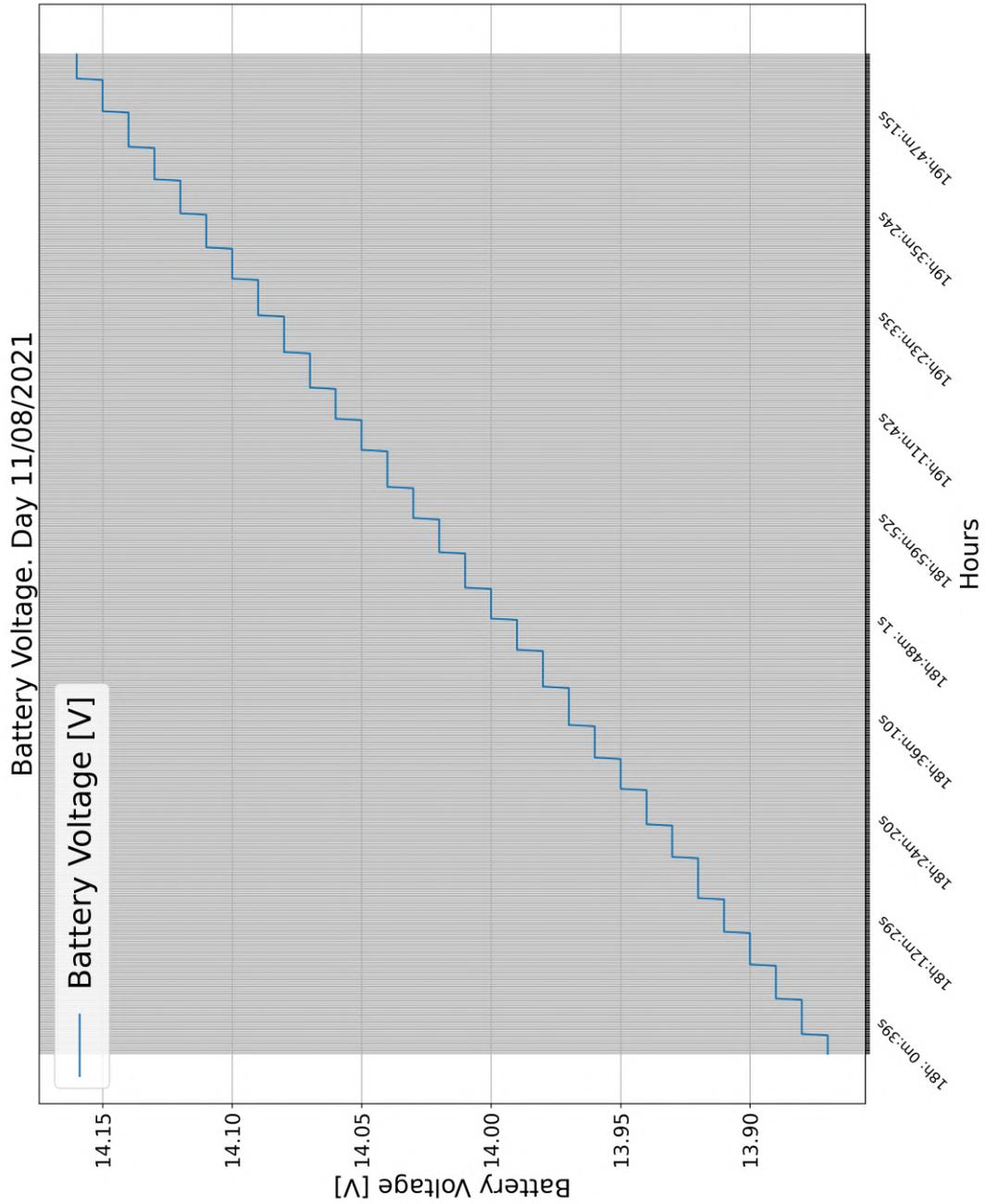


Figura D.20: SOC de la batería con *test* de 15 A.

**D.21. Tensión de la batería. *Test* de carga a 15 A**Figura D.21: Tensión de la batería con *test* de 15 A.

### D.22. Temperatura de la batería. Test de 10 A

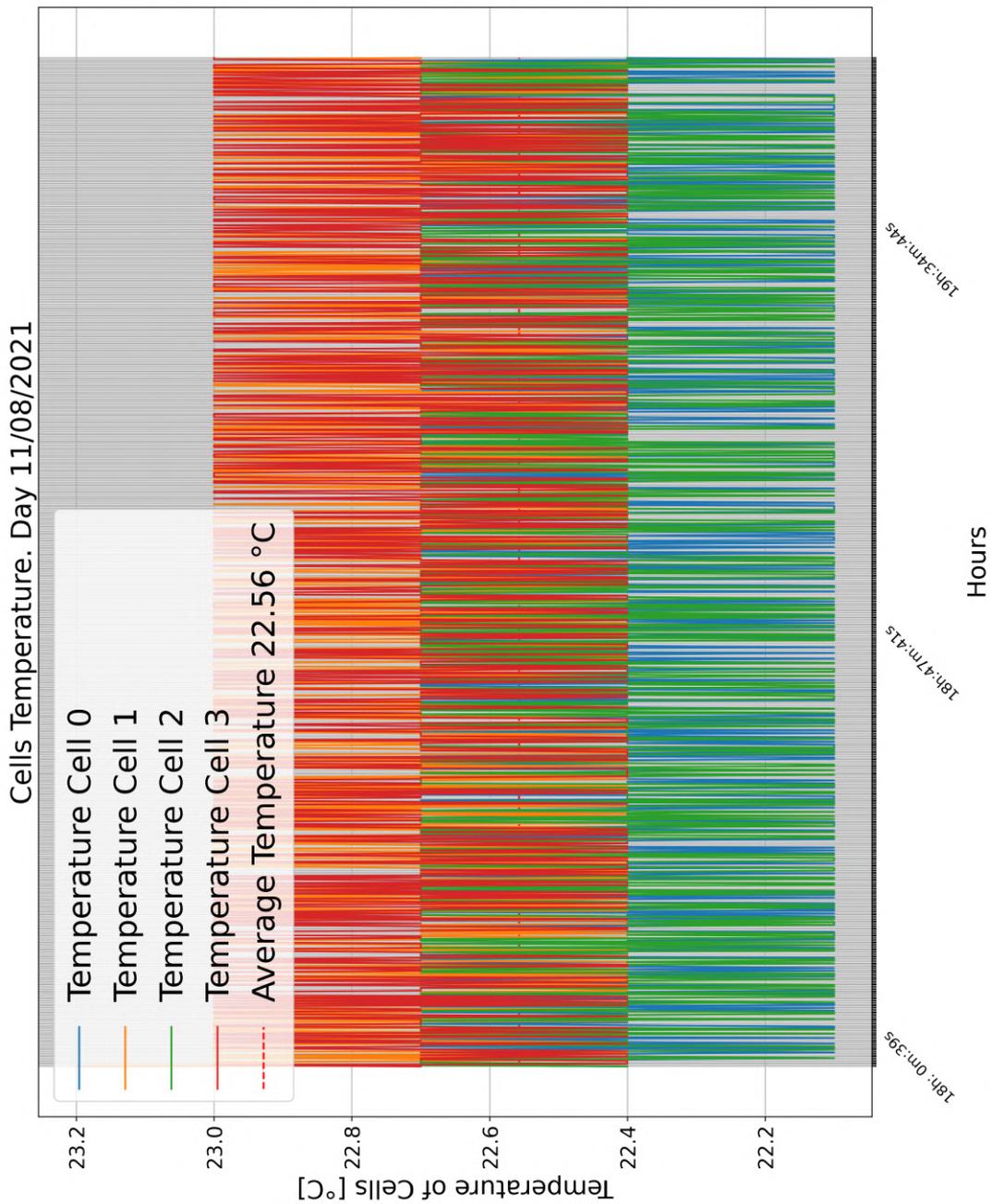


Figura D.22: Temperatura de la batería con test de 15 A.



## **Apéndice E**

### ***Datasheets* de los Integrados y componentes utilizados**

Las siguientes hojas de características pertenecen a los distintos circuitos integrados y componentes que se han empleado en el circuito.

## E.1. LTC6804



### LTC6804-1/LTC6804-2 Multicell Battery Monitors

#### FEATURES

- Measures Up to 12 Battery Cells in Series
- Stackable Architecture Supports 100s of Cells
- Built-In isoSPI™ Interface:
  - 1Mbps Isolated Serial Communications
  - Uses a Single Twisted Pair, Up to 100 Meters
  - Low EMI Susceptibility and Emissions
- 1.2mV Maximum Total Measurement Error
- 290µs to Measure All Cells in a System
- Synchronized Voltage and Current Measurement
- 16-Bit Delta-Sigma ADC with Frequency Programmable 3rd Order Noise Filter
- Engineered for ISO26262 Compliant Systems
- Passive Cell Balancing with Programmable Timer
- 5 General Purpose Digital I/O or Analog Inputs:
  - Temperature or other Sensor Inputs
  - Configurable as an I<sup>2</sup>C or SPI Master
- 4µA Sleep Mode Supply Current
- 48-Lead SSOP Package

#### APPLICATIONS

- Electric and Hybrid Electric Vehicles
- Backup Battery Systems
- Grid Energy Storage
- High Power Portable Equipment

#### DESCRIPTION

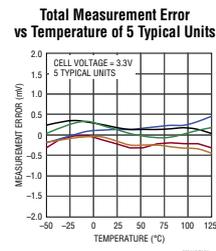
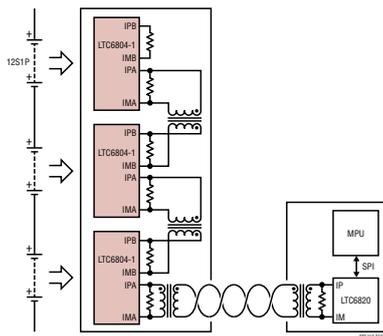
The LTC6804 is a 3rd generation multicell battery stack monitor that measures up to 12 series connected battery cells with a total measurement error of less than 1.2mV. The cell measurement range of 0V to 5V makes the LTC6804 suitable for most battery chemistries. All 12 cell voltages can be captured in 290µs, and lower data acquisition rates can be selected for high noise reduction.

Multiple LTC6804 devices can be connected in series, permitting simultaneous cell monitoring of long, high voltage battery strings. Each LTC6804 has an isoSPI interface for high speed, RF-immune, local area communications. Using the LTC6804-1, multiple devices are connected in a daisy-chain with one host processor connection for all devices. Using the LTC6804-2, multiple devices are connected in parallel to the host processor, with each device individually addressed.

Additional features include passive balancing for each cell, an onboard 5V regulator, and 5 general purpose I/O lines. In sleep mode, current consumption is reduced to 4µA. The LTC6804 can be powered directly from the battery, or from an isolated supply.

LT, LT, LTC, LTM, Linear Technology and the Linear logo are registered and isoSPI is a trademark of Linear Technology Corporation. All other trademarks are the property of their respective owners. Protected by U.S. patents, including 8908799, 9182428, 9270133.

#### TYPICAL APPLICATION



6804121c



For more information [www.linear.com/LTC6804-1](http://www.linear.com/LTC6804-1)

## E.2. ACS712



### ACS712

**Fully Integrated, Hall Effect-Based Linear Current Sensor  
with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor**

**Features and Benefits**

- Low-noise analog signal path
- Device bandwidth is set via the new FILTER pin
- 5  $\mu$ s output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at  $T_A = 25^\circ\text{C}$
- Small footprint, low-profile SOIC8 package
- 1.2 m $\Omega$  internal conductor resistance
- 2.1 kV<sub>RMS</sub> minimum isolation voltage from pins 1-4 to pins 5-8
- 5.0 V, single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis
- Ratiometric output from supply voltage



**Package: 8 Lead SOIC (suffix LC)**



Approximate Scale 1:1

**Description**

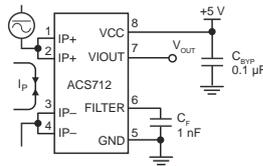
The Allegro® ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switched-mode power supplies, and overcurrent fault protection.

The device consists of a precise, low-offset, linear Hall sensor circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which is sensed by the integrated Hall IC and converted into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging.

The output of the device has a positive slope ( $>V_{IOUT(Q)}$ ) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sensing. The internal resistance of this conductive path is 1.2 m $\Omega$  typical, providing low power

*Continued on the next page...*

**Typical Application**



Application 1. The ACS712 outputs an analog signal,  $V_{OUT}$ , that varies linearly with the uni- or bi-directional AC or DC primary sensed current,  $I_p$ , within the range specified.  $C_F$  is recommended for noise management, with values that depend on the application.

## E.3. BSS308PE



BSS308PE

### OptiMOS™ P3 Small-Signal-Transistor

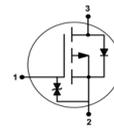
#### Features

- P-channel
- Enhancement mode
- Logic level (4.5V rated)
- ESD protected
- Qualified according to AEC Q101
- 100% lead-free; RoHS compliant
- Halogen-free according to IEC61249-2-21

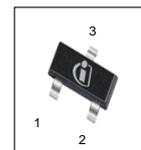


#### Product Summary

$V_{DS}$		-30	V
$R_{DS(on),max}$	$V_{GS}=-10\text{ V}$	80	m $\Omega$
	$V_{GS}=-4.5\text{ V}$	130	
$I_D$		-2.0	A



PG-SOT-23



Type	Package	Tape and Reel Information	Marking	Lead Free	Packing
BSS308PE	PG-SOT23	H6327: 3000 pcs/ reel	YFs	Yes	Non dry

Maximum ratings, at  $T_j=25\text{ °C}$ , unless otherwise specified

Parameter	Symbol	Conditions	Value	Unit
Continuous drain current	$I_D$	$T_A=25\text{ °C}$	-2.0	A
		$T_A=70\text{ °C}$	-1.6	
Pulsed drain current	$I_{D,pulse}$	$T_A=25\text{ °C}$	-8.0	
Avalanche energy, single pulse	$E_{AS}$	$I_D=-2\text{ A}$ , $R_{GS}=25\text{ }\Omega$	-10.7	mJ
Reverse diode dv/dt	dv/dt	$I_D=-2\text{ A}$ , $V_{DS}=-16\text{ V}$ , $di/dt=-200\text{ A}/\mu\text{s}$ , $T_{j,max}=150\text{ °C}$	6	kV/ $\mu\text{s}$
Gate source voltage	$V_{GS}$		$\pm 20$	V
Power dissipation <sup>1)</sup>	$P_{tot}$	$T_A=25\text{ °C}$	0.5	W
Operating and storage temperature	$T_j$ , $T_{stg}$		-55 ... 150	°C
ESD Class		JESD22-A114 -HBM	2 (2kV to 4kV)	
Soldering Temperature			260 °C	°C
IEC climatic category; DIN IEC 68-1			55/150/56	°C

## E.4. ADG731



16-/32-Channel, Serially Controlled 4  $\Omega$   
1.8 V to 5.5 V,  $\pm 2.5$  V, Analog Multiplexers

ADG725/ADG731

### FEATURES

- 3-Wire SPI Compatible Serial Interface
- 1.8 V to 5.5 V Single Supply
- $\pm 2.5$  V Dual-Supply Operation
- 4  $\Omega$  On Resistance
- 0.5  $\Omega$  On Resistance Flatness
- 7 mm x 7 mm 48-Lead Chip Scale Package (LFCSP)  
or 48-Lead TQFP Package
- Rail-to-Rail Operation
- Power-On Reset
- 42 ns Switching Times
- Single 32-to-1 Channel Multiplexer
- Dual/Differential 16-to-1 Channel Multiplexer
- TTL/CMOS Compatible Inputs
- For Functionally Equivalent Devices with Parallel  
Interface, See ADG726/ADG732

### APPLICATIONS

- Optical Applications
- Data Acquisition Systems
- Communication Systems
- Relay Replacement
- Audio and Video Switching
- Battery-Powered Systems
- Medical Instrumentation
- Automatic Test Equipment

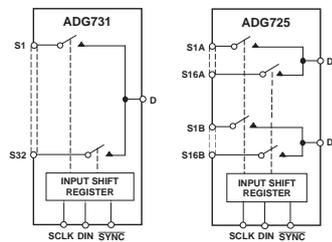
### GENERAL DESCRIPTION

The ADG731/ADG725 are monolithic, CMOS, 32-channel/dual 16-channel analog multiplexers with a serially controlled 3-wire interface. The ADG731 switches one of 32 inputs (S1-S32) to a common output, D. The ADG725 can be configured as a dual mux switching one of 16 inputs to one output, or a differential mux switching one of 16 inputs to a differential output.

These multiplexers utilize a 3-wire serial interface that is compatible with SPI<sup>®</sup>, QSPI<sup>™</sup>, MICROWIRE<sup>™</sup>, and some DSP interface standards. On power-up, the Internal Shift Register contains all zeros and all switches are in the OFF state.

These multiplexers are designed on an enhanced submicron process that provides low power dissipation yet gives high switching speed with very low on resistance and leakage currents. They operate from a single supply of 1.8 V to 5.5 V or a  $\pm 2.5$  V dual supply, making them ideally suited to a variety of applications. On resistance is in the region of a few ohms, is closely matched between switches, and is very flat over the full signal range.

### FUNCTIONAL BLOCK DIAGRAM



These parts can operate equally well as either multiplexers or demultiplexers and have an input signal range that extends to the supplies. In the OFF condition, signal levels up to the supplies are blocked. All channels exhibit break-before-make switching action, preventing momentary shorting when switching channels.

The ADG731 and ADG725 are serially controlled 32-channel, and dual/differential 16-channel multiplexers, respectively. They are available in either a 48-lead LFCSP or TQFP package.

### PRODUCT HIGHLIGHTS

1. 3-Wire Serial Interface.
2. 1.8 V to 5.5 V Single-Supply or  $\pm 2.5$  V Dual-Supply Operation. These parts are specified and guaranteed with 5 V  $\pm 10\%$ , 3 V  $\pm 10\%$  single-supply, and  $\pm 2.5$  V  $\pm 10\%$  dual-supply rails.
3. On Resistance of 4  $\Omega$ .
4. Guaranteed Break-Before-Make Switching Action.
5. 7 mm x 7 mm 48-Lead Chip Scale Package (LFCSP) or 48-Lead TQFP Package.

Rev. B Document Feedback  
Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
Tel: 781.329.4700 ©2015 Analog Devices, Inc. All rights reserved.  
Technical Support [www.analog.com](http://www.analog.com)

## E.5. ATmega 328P



8-bit AVR Microcontrollers

ATmega328/P

DATASHEET SUMMARY

### Introduction

The Atmel<sup>®</sup> picoPower<sup>®</sup> ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR<sup>®</sup> enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

### Feature

High Performance, Low Power Atmel<sup>®</sup>AVR<sup>®</sup> 8-Bit Microcontroller Family

- Advanced RISC Architecture
  - 131 Powerful Instructions
  - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 32KBytes of In-System Self-Programmable Flash program Memory
  - 1KBytes EEPROM
  - 2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data Retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Atmel<sup>®</sup> QTouch<sup>®</sup> Library Support
  - Capacitive Touch Buttons, Sliders and Wheels
  - QTouch and QMatrix<sup>®</sup> Acquisition
  - Up to 64 sense channels

# E.6. MAX232

Click [here](#) for production status of specific part numbers.

## MAX220–MAX249

## +5V-Powered, Multichannel RS-232 Drivers/Receivers

### General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where  $\pm 12V$  is not available.

The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

The MAX220–MAX249 are offered in 26 different packages with temperatures from 0 to +70°C up to -55°C to +125°C. See ordering information table at the end of the data sheet for all package and temperature options.

### Applications

- Interface Translation
- Multidrop RS-232 Networks
- Portable Diagnostics Equipment

### Benefits and Features

Saves Board Space

- Integrated Charge Pump Circuitry
  - Eliminates the Need for a Bipolar  $\pm 12V$  Supply
  - Enables Single Supply Operation from +5V Supply
- Integrated Capacitors (MAX223, MAX233, MAX235, MAX245–MAX247)

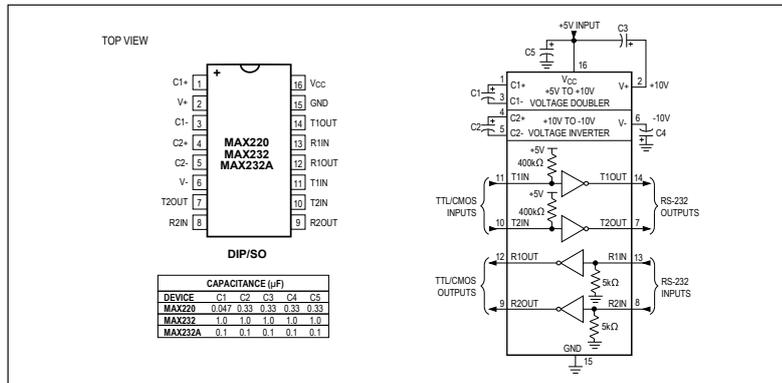
Saves Power for Reduced Power Requirements

- 5 $\mu W$  Shutdown Mode

*Ordering Information and Selection Table appears at end of data sheet.*

*AutoShutdown and UCSP are trademarks of Maxim Integrated Products, Inc.*

### MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit



## E.7. MAX485



maxim  
integrated™

**MAX481E/MAX483E/MAX485E/  
MAX487E-MAX491E/MAX1487E**

### **±15kV ESD-Protected, Slew-Rate-Limited, Low-Power, RS-485/RS-422 Transceivers**

#### **General Description**

The MAX481E, MAX483E, MAX485E, MAX487E-MAX491E, and MAX1487E are low-power transceivers for RS-485 and RS-422 communications in harsh environments. Each driver output and receiver input is protected against ±15kV electro-static discharge (ESD) shocks, without latching. These parts contain one driver and one receiver. The MAX483E, MAX487E, MAX488E, and MAX489E feature reduced slew-rate drivers that minimize EMI and reduce reflections caused by improperly terminated cables, thus allowing error-free data transmission up to 250kbps. The driver slew rates of the MAX481E, MAX485E, MAX490E, MAX491E, and MAX1487E are not limited, allowing them to transmit up to 2.5Mbps.

These transceivers draw as little as 120µA supply current when unloaded or when fully loaded with disabled drivers (see *Selector Guide*). Additionally, the MAX481E, MAX483E, and MAX487E have a low-current shutdown mode in which they consume only 0.5µA. All parts operate from a single +5V supply.

Drivers are short-circuit current limited, and are protected against excessive power dissipation by thermal shutdown circuitry that places their outputs into a high-impedance state. The receiver input has a fail-safe feature that guarantees a logic-high output if the input is open circuit.

The MAX487E and MAX1487E feature quarter-unit-load receiver input impedance, allowing up to 128 transceivers on the bus. The MAX488E-MAX491E are designed for full-duplex communications, while the MAX481E, MAX483E, MAX485E, MAX487E, and MAX1487E are designed for half-duplex applications. For applications that are not ESD sensitive see the pin- and function-compatible MAX481, MAX483, MAX485, MAX487-MAX491, and MAX1487.

#### **Applications**

Low-Power RS-485 Transceivers  
Low-Power RS-422 Transceivers  
Level Translators  
Transceivers for EMI-Sensitive Applications  
Industrial-Control Local Area Networks

#### **Next-Generation Device Features**

- ◆ **For Fault-Tolerant Applications:**  
MAX3430: ±80V Fault-Protected, Fail-Safe, 1/4-Unit Load, +3.3V, RS-485 Transceiver  
MAX3080-MAX3089: Fail-Safe, High-Speed (10Mbps), Slew-Rate-Limited, RS-485/RS-422 Transceivers
- ◆ **For Space-Constrained Applications:**  
MAX3460-MAX3464: +5V, Fail-Safe, 20Mbps, Profibus, RS-485/RS-422 Transceivers  
MAX3362: +3.3V, High-Speed, RS-485/RS-422 Transceiver in a SOT23 Package  
MAX3280E-MAX3284E: ±15kV ESD-Protected, 52Mbps, +3V to +5.5V, SOT23, RS-485/RS-422 True Fail-Safe Receivers  
MAX3030E-MAX3033E: ±15kV ESD-Protected, +3.3V, Quad RS-422 Transmitters
- ◆ **For Multiple Transceiver Applications:**  
MAX3293/MAX3294/MAX3295: 20Mbps, +3.3V, SOT23, RS-485/RS-422 Transmitters
- ◆ **For Fail-Safe Applications:**  
MAX3440E-MAX3444E: ±15kV ESD-Protected, ±60V Fault-Protected, 10Mbps, Fail-Safe RS-485/J1708 Transceivers
- ◆ **For Low-Voltage Applications:**  
MAX3483E/MAX3485E/MAX3486E/MAX3488E/MAX3490E/MAX3491E: +3.3V Powered, ±15kV ESD-Protected, 12Mbps, Slew-Rate-Limited, True RS-485/RS-422 Transceivers

#### **Ordering Information**

PART	TEMP RANGE	PIN-PACKAGE
MAX481ECPA	0°C to +70°C	8 Plastic DIP
MAX481ECSA	0°C to +70°C	8 SO
MAX481ECPA	-40°C to +85°C	8 Plastic DIP
MAX481EESA	-40°C to +85°C	8 SO
MAX483ECPA	0°C to +70°C	8 Plastic DIP
MAX483ECSA	0°C to +70°C	8 SO
MAX483ECPA	-40°C to +85°C	8 Plastic DIP
MAX483EESA	-40°C to +85°C	8 SO

Ordering Information continued at end of data sheet.

Selector Guide appears at end of data sheet.

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at [www.maximintegrated.com](http://www.maximintegrated.com).

19-0410; Rev 4; 10/03

# E.8. MCP2515



# MCP2515

## Stand-Alone CAN Controller with SPI Interface

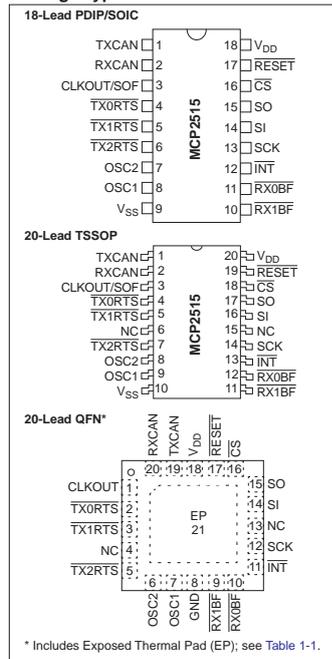
### Features

- Implements CAN V2.0B at 1 Mb/s:
  - 0 to 8-byte length in the data field
  - Standard and extended data and remote frames
- Receive Buffers, Masks and Filters:
  - Two receive buffers with prioritized message storage
  - Six 29-bit filters
  - Two 29-bit masks
- Data Byte Filtering on the First Two Data Bytes (applies to standard data frames)
- Three Transmit Buffers with Prioritization and Abort Features
- High-Speed SPI Interface (10 MHz):
  - SPI modes 0,0 and 1,1
- One-Shot mode Ensures Message Transmission is Attempted Only One Time
- Clock Out Pin with Programmable Prescaler:
  - Can be used as a clock source for other device(s)
- Start-of-Frame (SOF) Signal is Available for Monitoring the SOF Signal:
  - Can be used for time slot-based protocols and/or bus diagnostics to detect early bus degradation
- Interrupt Output Pin with Selectable Enables
- Buffer Full Output Pins Configurable as:
  - Interrupt output for each receive buffer
  - General purpose output
- Request-to-Send (RTS) Input Pins Individually Configurable as:
  - Control pins to request transmission for each transmit buffer
  - General purpose inputs
- Low-Power CMOS Technology:
  - Operates from 2.7V-5.5V
  - 5 mA active current (typical)
  - 1 µA standby current (typical) (Sleep mode)
- Temperature Ranges Supported:
  - Industrial (I): -40°C to +85°C
  - Extended (E): -40°C to +125°C

### Description

Microchip Technology's MCP2515 is a stand-alone Controller Area Network (CAN) controller that implements the CAN specification, Version 2.0B. It is capable of transmitting and receiving both standard and extended data and remote frames. The MCP2515 has two acceptance masks and six acceptance filters that are used to filter out unwanted messages, thereby reducing the host MCU's overhead. The MCP2515 interfaces with microcontrollers (MCUs) via an industry standard Serial Peripheral Interface (SPI).

### Package Types



## E.9. TJA1050T

Philips Semiconductors

Product specification

### High speed CAN transceiver

TJA1050

#### FEATURES

- Fully compatible with the "ISO 11898" standard
- High speed (up to 1 Mbaud)
- Very low ElectroMagnetic Emission (EME)
- Differential receiver with wide common-mode range for high ElectroMagnetic Immunity (EMI)
- An unpowered node does not disturb the bus lines
- Transmit Data (TXD) dominant time-out function
- Silent mode in which the transmitter is disabled
- Bus pins protected against transients in an automotive environment
- Input levels compatible with 3.3 V and 5 V devices
- Thermally protected
- Short-circuit proof to battery and to ground
- At least 110 nodes can be connected.

#### GENERAL DESCRIPTION

The TJA1050 is the interface between the Controller Area Network (CAN) protocol controller and the physical bus. The device provides differential transmit capability to the bus and differential receive capability to the CAN controller.

The TJA1050 is the third Philips high-speed CAN transceiver after the PCA82C250 and the PCA82C251. The most important differences are:

- Much lower electromagnetic emission due to optimal matching of the output signals CANH and CANL
- Improved behaviour in case of an unpowered node
- No standby mode.

This makes the TJA1050 eminently suitable for use in nodes that are in a power-down situation in partially powered networks.

#### QUICK REFERENCE DATA

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
$V_{CC}$	supply voltage		4.75	5.25	V
$V_{CANH}$	DC voltage at pin CANH	$0 < V_{CC} < 5.25$ V; no time limit	-27	+40	V
$V_{CANL}$	DC voltage at pin CANL	$0 < V_{CC} < 5.25$ V; no time limit	-27	+40	V
$V_{(dif)(bus)}$	differential bus input voltage	dominant	1.5	3	V
$t_{PD(TXD-RXD)}$	propagation delay TXD to RXD	$V_S = 0$ V; see Fig.7	-	250	ns
$T_{vj}$	virtual junction temperature		-40	+150	°C

#### ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
TJA1050T	SO8	plastic small outline package; 8 leads; body width 3.9 mm	SOT96-1
TJA1050U	-	bare die; die dimensions 1700 × 1280 × 380 μm	-

## E.10. HC-49 Oscilador

### CRYSTAL UNIT HC-49US DIP



#### Applications Features

- Industrial Control Consumers.
- Dimensions:11.5 x 4.50 x3.68 mm.
- Through hole type crystal units.
- Frequency range:3.2768~64MHz
- A great number of standard frequencies.

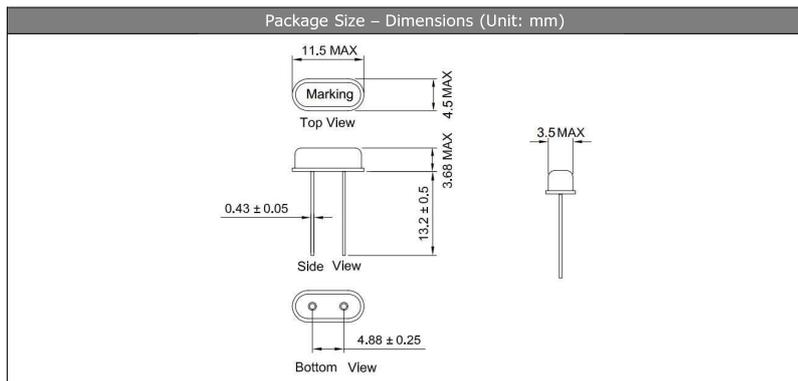
#### Specifications

Standard Frequency	3.2768~64 MHz
Vibration Mode	AT Fundamental
Load Capacitance	12pF,20pF,or specify
Frequency Tolerance (at 25 °C)	±10ppm,±20ppm,or specify
Frequency Versus Temperature Characteristics	±20ppm,or specify
Operating Temperature	-20~+70°C, -40~+85°C, or specify
Storage Temperature	-40~+85°C or specify
Shunt Capacitance	7 pF Max.
Level of Drive	1~500µWMax.(100uW typical)
Aging (at 25 °C )	±3ppm/year Max.

#### Equivalent Series Resistance(ESR)

Fundamental			
3.2768 ~ 4MHz	180 Ω Max.	6 ~ 17 MHz	80 Ω Max.
4 ~ 5MHz	120 Ω Max.	7 ~ 10 MHz	60 Ω Max.
5 ~ 6 MHz	100 Ω Max.	10 ~ 27MHz	40 Ω Max.
3rd Overtone			
20 ~ 25 MHz	100 Ω Max.	25 ~ 64 MHz	80 Ω Max.

#### Dimensions and Patterns [unit:mm]



# E.11. INA 219

-  Product Folder
-  Sample & Buy
-  Technical Documents
-  Tools & Software
-  Support & Community



INA219

SBOS448G –AUGUST 2008 –REVISED DECEMBER 2015

## INA219 Zero-Drift, Bidirectional Current/Power Monitor With I<sup>2</sup>C Interface

### 1 Features

- Senses Bus Voltages from 0 to 26 V
- Reports Current, Voltage, and Power
- 16 Programmable Addresses
- High Accuracy: 0.5% (Maximum) Over Temperature (INA219B)
- Filtering Options
- Calibration Registers
- SOT23-8 and SOIC-8 Packages

### 2 Applications

- Servers
- Telecom Equipment
- Notebook Computers
- Power Management
- Battery Chargers
- Welding Equipment
- Power Supplies
- Test Equipment

### 3 Description

The INA219 is a current shunt and power monitor with an I<sup>2</sup>C- or SMBUS-compatible interface. The device monitors both shunt voltage drop and bus supply voltage, with programmable conversion times and filtering. A programmable calibration value, combined with an internal multiplier, enables direct readouts of current in amperes. An additional multiplying register calculates power in watts. The I<sup>2</sup>C- or SMBUS-compatible interface features 16 programmable addresses.

The INA219 is available in two grades: A and B. The B grade version has higher accuracy and higher precision specifications.

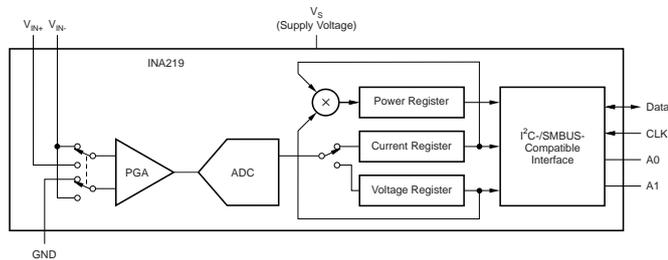
The INA219 senses across shunts on buses that can vary from 0 to 26 V. The device uses a single 3- to 5.5-V supply, drawing a maximum of 1 mA of supply current. The INA219 operates from -40°C to 125°C.

#### Device Information<sup>(1)</sup>

PART NUMBER	PACKAGE	BODY SIZE (NOM)
INA219	SOIC (8)	3.91 mm x 4.90 mm
	SOT-23 (8)	1.63 mm x 2.90 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

#### Simplified Schematic



 An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

## E.12. MP 2307 DC/DC



### MP2307 3A, 23V, 340KHz Synchronous Rectified Step-Down Converter

#### DESCRIPTION

The MP2307 is a monolithic synchronous buck regulator. The device integrates 100mΩ MOSFETs that provide 3A of continuous load current over a wide operating input voltage of 4.75V to 23V. Current mode control provides fast transient response and cycle-by-cycle current limit.

An adjustable soft-start prevents inrush current at turn-on and in shutdown mode, the supply current drops below 1μA.

This device, available in an 8-pin SOIC package, provides a very compact system solution with minimal reliance on external components.

#### EVALUATION BOARD REFERENCE

Board Number	Dimensions
EV2307DN-00A	2.0"X x 1.5"Y x 0.5"Z

#### FEATURES

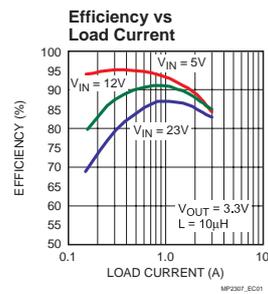
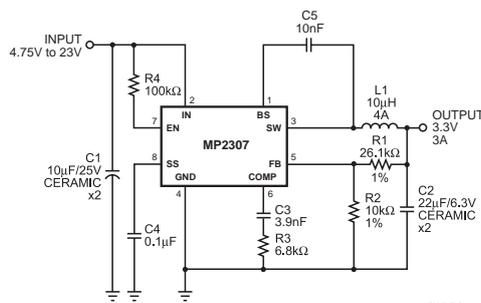
- 3A Continuous Output Current
- 4A Peak Output Current
- Wide 4.75V to 23V Operating Input Range
- Integrated 100mΩ Power MOSFET Switches
- Output Adjustable from 0.925V to 20V
- Up to 95% Efficiency
- Programmable Soft-Start
- Stable with Low ESR Ceramic Output Capacitors
- Fixed 340KHz Frequency
- Cycle-by-Cycle Over Current Protection
- Input Under Voltage Lockout
- Thermally Enhanced 8-Pin SOIC Package

#### APPLICATIONS

- Distributed Power Systems
- Networking Systems
- FPGA, DSP, ASIC Power Supplies
- Green Electronics/Appliances
- Notebook Computers

\*MPS and "The Future of Analog IC Technology" are Registered Trademarks of Monolithic Power Systems, Inc.

#### TYPICAL APPLICATION



## E.13. Resistencia 10Ω

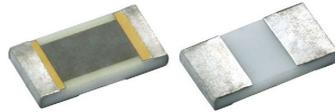


www.vishay.com

PHP

Vishay Dale Thin Film

### High Power Thin Film Wraparound Chip Resistor



#### ADDITIONAL RESOURCES



PHP series chip resistors are designed with enlarged backside terminations to reduce the thermal resistance between the topside resistor layer and the solder joint on the end users circuit board.

Actual power handling capability is limited by the end user mounting process. As with any high power chip resistor the ability to remove the generated heat is critical to the overall performance of the device.

#### FEATURES

- High purity ceramic substrate
- Power rating to 2.5 W
- Resistance range 10 Ω to 30.1 kΩ
- Resistor tolerance to ± 0.1 %
- TCR to ± 25 ppm/°C
- Flame resistant UL 94 V-0

#### APPLICATIONS

- Power supplies
- Power switching
- Braking system
- Test and measurement equipment
- Motor deflection circuits

#### TYPICAL PERFORMANCE

	ABSOLUTE
TCR	25
TOL.	0.1

STANDARD ELECTRICAL SPECIFICATIONS		
TEST	SPECIFICATIONS	CONDITIONS
Material	Nichrome	
Resistance Range	10 Ω to 30.1 kΩ	-
TCR: Absolute	25 ppm/°C, 50 ppm/°C (standard) and, 100 ppm/°C	-55 °C to +125 °C
Tolerance: Absolute	0.1 %, 0.5 %, 1.0 % and, 5.0 %	+25 °C
Power Rating: Resistor	0.375 W - 2.5 W <sup>(1)</sup>	Maximum at +70 °C
Stability: Absolute	ΔR 0.1 %	2000 h at +70 °C
Stability: Ratio	Not applicable	-
Voltage Coefficient	< 0.1 ppm/V	-
Working Voltage	75 V to 200 V	-
Operating Temperature Range	-55 °C to +155 °C	-
Storage Temperature Range	-55 °C to +155 °C	-
Noise	< -30 dB	-
Shelf Life Stability: Absolute	± 0.01 %	1 year at +25 °C

COMPONENT RATINGS			
CASE SIZE	POWER RATING (mW)	WORKING VOLTAGE (V)	RESISTANCE RANGE (Ω)
0603	375 <sup>(1)</sup>	75	10 to 30.1K
0805	625 <sup>(1)</sup>	100	10 to 30.1K
1206	1000 <sup>(1)</sup>	200	10 to 30.1K
2512	2500 <sup>(1)</sup>	200	10 to 30.1K

Note

<sup>(1)</sup> Dependent on component mounting by user

ENVIRONMENTAL TESTS (Vishay Performance vs. MIL-PRF-55342 Requirements)		
ENVIRONMENTAL TEST	LIMITS MIL-PRF-55342 CHARACTERISTIC "E"	TYPICAL VISHAY PERFORMANCE
Resistance Temperature Characteristic	± 25 ppm/°C	± 15 ppm/°C
Maximum Ambient Temperature at Rated Wattage	+70 °C	+70 °C
Maximum Ambient Temperature at Power Derating	+150 °C	+150 °C
Thermal Shock	± 0.1 %	± 0.04 %
Low Temperature Operation	± 0.1 %	± 0.001 %
Short Time Overload	± 0.1 %	± 0.003 %
High Temperature Exposure	± 0.1 %	± 0.030 %
Resistance to Soldering Heat	± 0.2 %	± 0.007 %
Moisture Resistance	± 0.2 %	± 0.002 %
Life at +70 °C for 2000 h	± 0.5 %	± 0.100 %

Revision: 23-Oct-2019

1

Document Number: 60076

For technical questions, contact: [thinfilm@vishay.com](mailto:thinfilm@vishay.com)

THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT [www.vishay.com/doc/291000](http://www.vishay.com/doc/291000)

## E.14. Conector RJ45

Ethernet Connectors



### FEATURES

- Suitable for Cat5e Ethernet Applications
- Compatible with cable mount RJ45 plugs
- Right Angle Connection
- Gold over nickel contact plating for a reliable connection

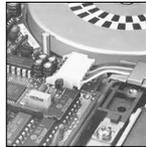
### RS PRO, Female RJ45 Connector

RS Stock No.: 331-6443



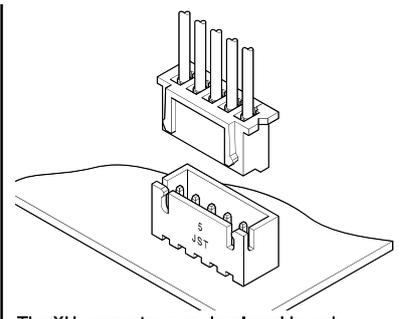
RS Professionally Approved Products bring to you professional quality parts across all product categories. Our product range has been tested by engineers and provides a comparable quality to the leading brands without paying a premium price.

# E.15. Conector JST-XH 14



## XH CONNECTOR

2.5 mm pitch/Disconnectable Crimp style connectors



The XH connector was developed based on the high reliability and versatility of our NH series connectors. The connector is very small with a mounting height of 9.8 mm.

- Original folded beam contact
- Box-shaped shrouded header
- Header with a boss
- Interchangeability
- Conforming to the HA terminal

### Specifications

- Current rating: 3 A AC/DC (AWG #22)
- Voltage rating: 250 V AC/DC
- Temperature range: -25°C to +85°C  
(including temperature rise in applying electrical current)
- Contact resistance: Initial value/ 10 mΩ max.  
After environmental tests/ 20 mΩ max.
- Insulation resistance: 1,000 MΩ min.
- Withstanding voltage: 1,000 VAC/minute
- Applicable wire: AWG #30 to #22
- Applicable PC board thickness: 1.6 mm

\* In using the products, refer to "Handling Precautions for Terminals and Connectors" described on our website (Technical documents of Product information page).

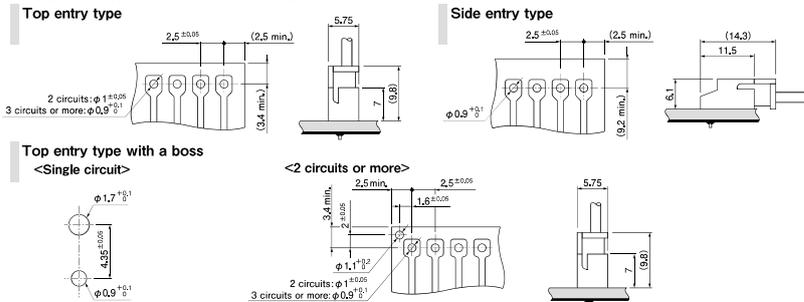
- \* RoHS2 compliance
- \* Dimensional unit: mm
- \* Contact JST for details.

### Standards

- Recognized E60389
- J50014297

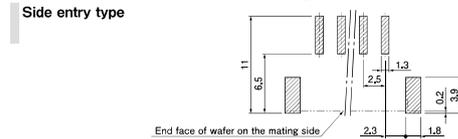
### PC board layout and Assembly layout

(Through-hole type (viewed from soldering side))



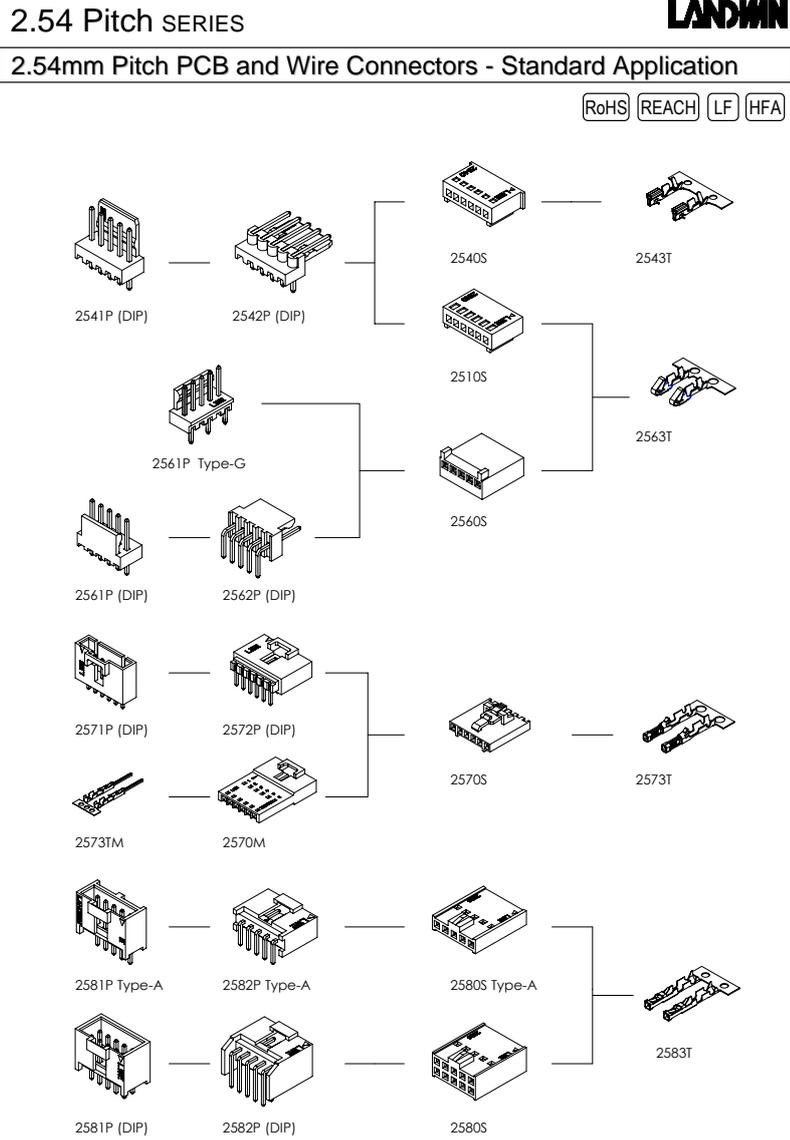
Note: 1. Tolerances are non-cumulative: ± 0.05 mm for all centers.  
2. Hole dimensions differ according to the type of PC board and piercing method. If PC boards made of hard material are used, the hole dimensions should be larger. The dimensions above should serve as a guideline. Contact JST for details.

(SMT type (viewed from connector mounting side))



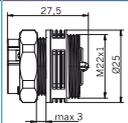
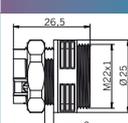
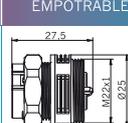
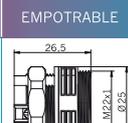
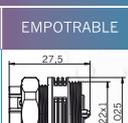
Note: 1. Tolerances are non-cumulative: ± 0.05 mm for all centers.  
2. The dimensions above should serve as a guideline. Contact JST for details.

## E.16. Conector KF2510 2 pines



## E.17. Conector IP68 WP3/EH

NUEVA

CONECTOR HEMBRA PARA MONTAJE EN PANEL IP68 2 POLOS + ⚡					
<b>WP3/EH</b>	● EMPOTRABLE			25 uds. 150x120x110 mm	📦 0,40 Kg
<b>CABLE FLEXIBLE</b>  14,15 mm 7,8 mm 0,5 -2,5 mm <sup>2</sup>	 27,5 M2x2,1 Ø2,5 max 3	 +0,5 18,5 0 max 3 -20,0 +0,5	 -40°C +85°C Intensidad 450V / 17,5A		
CONECTOR MACHO PARA MONTAJE EN PANEL IP68 4 POLOS					
<b>WP4/EM</b>	● EMPOTRABLE			25 uds. 150x120x110 mm	📦 0,44 Kg
<b>CABLE FLEXIBLE</b>  14,15 mm 7,8 mm 0,5 -2,5 mm <sup>2</sup>	 26,5 M2x2,1 Ø2,5 max 3	 +0,5 18,5 0 max 3 -20,0 +0,5	 -40°C +85°C Intensidad 450V / 17,5A		
CONECTOR HEMBRA PARA MONTAJE EN PANEL IP68 4 POLOS					
<b>WP4/EH</b>	● EMPOTRABLE			25 uds. 150x120x110 mm	📦 0,44 Kg
<b>CABLE FLEXIBLE</b>  14,15 mm 7,8 mm 0,5 -2,5 mm <sup>2</sup>	 27,5 M2x2,1 Ø2,5 max 3	 +0,5 18,5 0 max 3 -20,0 +0,5	 -40°C +85°C Intensidad 450V / 17,5A		
CONECTOR MACHO PARA MONTAJE EN PANEL IP68 4 POLOS + ⚡					
<b>WP5/EM</b>	● EMPOTRABLE			25 uds. 150x120x110 mm	📦 0,41- Kg
<b>CABLE FLEXIBLE</b>  14,15 mm 7,8 mm 0,5 -2,5 mm <sup>2</sup>	 26,5 M2x2,1 Ø2,5 max 3	 +0,5 18,5 0 max 3 -20,0 +0,5	 -40°C +85°C Intensidad 450V / 17,5A		
CONECTOR HEMBRA PARA MONTAJE EN PANEL IP68 4 POLOS + ⚡					
<b>WP5/EH</b>	● EMPOTRABLE			25 uds. 150x120x110 mm	📦 0,41 Kg
<b>CABLE FLEXIBLE</b>  14,15 mm 7,8 mm 0,5 -2,5 mm <sup>2</sup>	 27,5 M2x2,1 Ø2,5 max 3	 +0,5 18,5 0 max 3 -20,0 +0,5	 -40°C +85°C Intensidad 450V / 17,5A		

# E.18. Portafusibile 5mm x 20mm

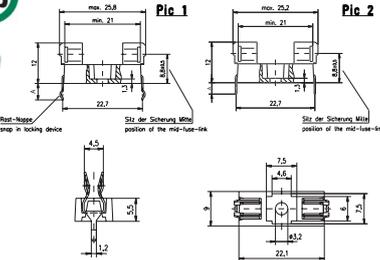
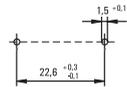
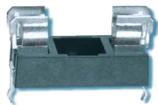
## MINIATURE FUSEHOLDERS



### Holder / No. 646

### PC Mount, for 5x20mm Fuses, lead free

### Specifications



**Mounting**  
Solder pins 0.5mm x 1.2mm  
Ø 3mm screw hole may be used optionally

**Materials**  
Holder: Black Thermoplastic, UL94 V-0  
Polyamide PA 6.6  
Metal Parts: Copper alloy, solderable, tinned

**Electrical Data (23°C)**  
Rated Voltage: 250V  
Max. current/power: 6.3A/2.5W  
5A/1W (with No. 648)  
6.3A/1.6W (with No. 640)

**Minimum Cross Section**  
Conducting path - 0.2mm<sup>2</sup>

**Unit Weight**  
1.6g

**Packaging**  
Bulk (1000 pcs.)

Order Information	Qty.	Order Number	Series	Type
			646	0000

Type	Pic	Meas. "A"	for PCB-Thickness
100	1	5mm	3mm
122	1	3.2mm	1.5mm
123	2	3.3mm	1.5mm

### Cover / No. 648

### Optional Cover



**Materials**  
Cover: Transparent Thermoplastic, UL94 V-0, Polycarbonate PC

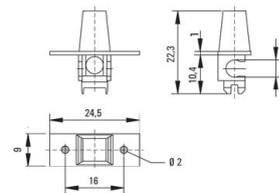
**Unit Weight**  
1.0g

**Packaging**  
Bulk (2000 pcs.)

Order Information	Qty.	Order Number	Series	Type
			648	0000 100

### Adapter / No. 640

### Optional Adapter



**Mounting**  
Hand adapter can be optionally used with Holder No. 646 and 647

**Materials**  
Adapter: Polyester PBT, UL94 V-0

**Unit Weight**  
1.0g

**Packaging**  
Bulk (5000 pcs.)

### Colours

Adapter black	100
Adapter brown	140
Adapter white	150
Adapter yellow	160
Adapter green	170
Adapter blue	190

Order Information	Qty.	Order Number	Series	Colour
			640	0000

Contact Littelfuse for an auto-insertable fuse, holder and adapter assembly.

Specifications are subject to change without notice

Note: 1.00 means the number one with two decimal places, 1,000 means the number one thousand.

## E.19. Raspberry Pi 3B+

Raspberry Pi 3 Model B+

2

### Specifications

<b>Processor:</b>	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
<b>Memory:</b>	1GB LPDDR2 SDRAM
<b>Connectivity:</b>	<ul style="list-style-type: none"> <li>■ 2.4GHz and 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2, BLE</li> <li>■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)</li> <li>■ 4 × USB 2.0 ports</li> </ul>
<b>Access:</b>	Extended 40-pin GPIO header
<b>Video &amp; sound:</b>	<ul style="list-style-type: none"> <li>■ 1 × full size HDMI</li> <li>■ MIPI DSI display port</li> <li>■ MIPI CSI camera port</li> <li>■ 4 pole stereo output and composite video port</li> </ul>
<b>Multimedia:</b>	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
<b>SD card support:</b>	Micro SD format for loading operating system and data storage
<b>Input power:</b>	<ul style="list-style-type: none"> <li>■ 5V/2.5A DC via micro USB connector</li> <li>■ 5V DC via GPIO header</li> <li>■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)</li> </ul>
<b>Environment:</b>	Operating temperature, 0–50°C
<b>Compliance:</b>	For a full list of local and regional product approvals, please visit <a href="http://www.raspberrypi.org/products/raspberry-pi-3-model-b+">www.raspberrypi.org/products/raspberry-pi-3-model-b+</a>
<b>Production lifetime:</b>	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.



# Apéndice F

## Diseño eléctrico de la batería

En el presente anexo se muestra la configuración adoptada para las celdas de la batería, así como su *datasheet*.

## F.1. Datasheet de la celda INR21700

-SAMSUNG SDI Confidential Proprietary -



Spec. No.	INR21700-40T	Version No.	0.0
-----------	--------------	-------------	-----

### 1.0. Scope

This product specification has been prepared to specify the rechargeable lithium-ion cell ('cell') to be supplied to Customer by Samsung SDI Co., Ltd.

### 2.0. Description and model

2.1 Description	lithium-ion rechargeable cell
2.2 Model name	INR21700-40T
2.3 Site	Manufactured in Korea

### 3.0. Nominal specifications (\*)

Item	Specification
3.1 Standard discharge capacity	Min. 4,000mAh - Charge: 0.5C(2A), 4.20V, 0.05C(200mA) cut-off @ RT - Discharge: 0.2C(800mA), 2.5V cut-off @ RT
3.2 Rated discharge capacity	Min. 3,900 mAh - Charge: 1.5C(6A), 4.20V, 0.025C(100mA) cut-off @ RT - Discharge: 10A, 2.5V cut-off @ RT
3.3 Nominal voltage	3.6V
3.4 Standard charge	CCCV, 2A, 4.20 V, 200mA cut-off
3.5 Rated charge	CCCV, 6A, 4.20 V, 100mA cut-off
3.6 Charging time	Standard charge : 180min / 200mA cut-off(@ RT) Rated charge: 70min / 100mA cut-off(@ RT)
3.7 Max. continuous discharge (Surface temperature)	35A(Without temperature cut) 45A(With 80 °C temperature cut)
3.8 Discharge cut-off voltage	2.5V (End of discharge)
3.9 Cycle life	Capacity $\geq$ 2,400mAh @ after 250cycles (60% of the standard capacity @ RT) - Charge : 6A, 4.20V, CCCV 100mA cut-off @ RT - Discharge: 35A , 2.5V cut-off @ RT
3.10 Retention characteristics	Capacity recovery(after the storage) $\geq$ 3,510 mAh (90% of the rated capacity @ RT) - Charge : 6A, 4.20V, CCCV 100mA cut-off @ RT - Storage : 30 days (@ 60 °C) - Discharge : 10A , 2.5V cut-off @ RT
3.11 Cell weight	70.0g max
3.12 Cell dimension	Height : Max 70.30mm Diameter : Max 21.22mm

## F.2. Chapa de Cobre de 0.8 mm

cobre

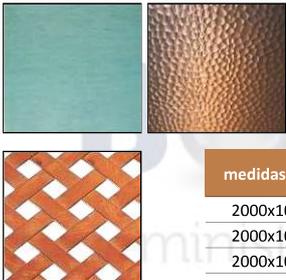
chapa

**Chapa CW024A R240**  
Acabado normal



medidas mm	espesor mm	peso	Ref.
2000x1000	0.6	10.74 K	12155500
2000x1000	0.8	14.30 K	12155520
2000x1000	1.0	17.88 K	12155530
2000x1000	1.5	26.82 K	12155540
2000x1000	2.0	35.80 K	12155550
2000x1000	3.0	54.20 K	12155560
2000x1000	4.0	71.60 K	12155570
2000x1000	5.0	90 K	12155580

**Chapa**  
Acabado patina, martillado y perforada



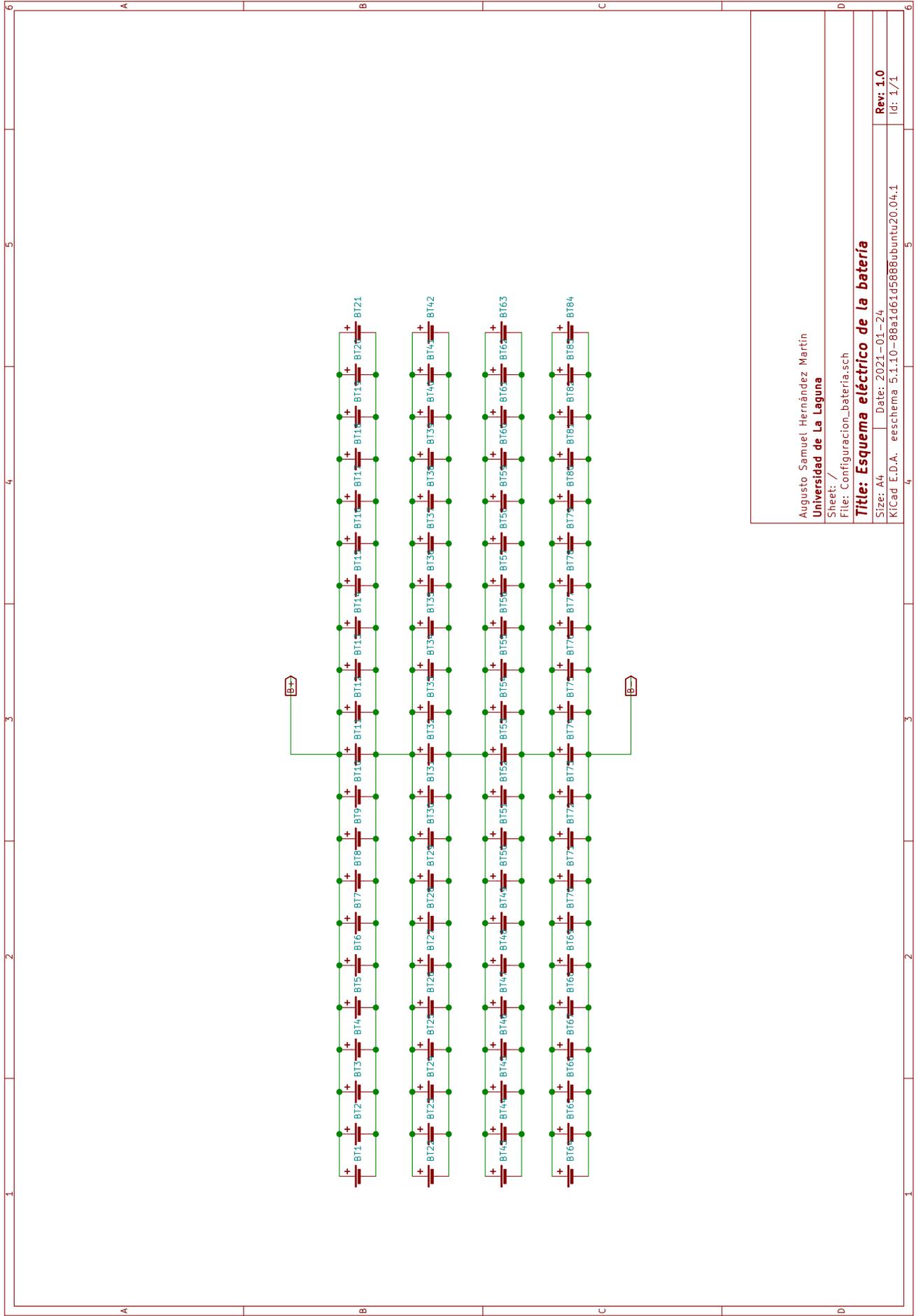
medidas mm	espesor mm	acabado	Ref.
2000x1000	0.7	patina	12155750
2000x1000	0.8	martillado	12155650
2000x1000	1.0	perforada cuadr. 20X20	12156615
2000x1000	1.5	perforada cuadr. 20X20	12156625



Pol. Industrial los Majuelos, Calle Puntallana nº6.  
C.P. 38108. La Laguna. Tenerife - Canarias, Spain  
Tfno: +34 922 824 140 - Fax: +34 922 824 320



Pol. Industrial Las Torres, Calle Arrecife nº 6.  
C.P.35010. Las Palmas de G. Canaria. Canarias, Spain.  
Tfno: +34 928 411 287 - Fax: +34 928 428 591



Augusto Samuel Hernández Martín  
**Universidad de La Laguna**  
 Sheet: /  
**Title: Esquema eléctrico de la batería**  
 Size: A4 Date: 2021-01-24  
 File: Configuración\_bateria.sch  
 KiCad E.D.A. eschema 5.1.10-88a1d61d5888@ubuntu20.04.1  
 Rev: 1.0  
 Id: 1/1

## **Apéndice G**

# **Dimensionado y simulación del generador fotovoltaico**

En este anexo se muestran los cálculos realizados para seleccionar el generador fotovoltaico así como los resultados de la posterior simulación en PVGIS [17].

# G.1. Cálculo del generador

Dimensionado Generador Fotovoltaico  
Cálculo de la inclinación de los paneles por el criterio del mes crítico

Radiación solar diaria superficie horizontal (Wh/m <sup>2</sup> /día)		Irradiación solar diaria (Wh/m <sup>2</sup> )											
		28°	29°	30°	32°	34°	36°	38°	40°	42°	44°	46°	
Enero	3290	2974	2986	2998	3021	3040	3056	3070	3081	3090	3095	3098	
Febrero	3900	4848	4866	4884	4915	4941	4963	4979	4991	4998	5000	4997	
Marzo	4840	5223	5226	5224	5217	5205	5188	5166	5139	5139	5107	5070	
Abril	5390	4676	4660	4644	4608	4568	4524	4477	4425	4370	4311	4249	
Mayo	5900	4942	4912	4882	4819	4751	4679	4603	4524	4440	4353	4263	
Junio	6390	5191	5152	5113	5030	4943	4852	4756	4657	4553	4446	4335	
Julio	5440	5975	5935	5893	5805	5712	5614	5512	5405	5293	5176	5054	
Agosto	4970	5210	5187	5163	5111	5055	4994	4928	4858	4784	4706	4623	
Septiembre	4280	5266	5256	5250	5233	5211	5184	5152	5116	5075	5029	4978	
Octubre	3490	5992	5106	5119	5141	5159	5171	5178	5180	5177	5168	5155	
Noviembre	3160	4933	4801	4697	4736	4781	4821	4855	4885	4910	4930	4945	
Diciembre		4970	4899	4839	4785	4765	4820	4871	4918	4960	4991	5021	

$L_{min} = (Wh) = 184,63$
Irradiación media (Wh/m <sup>2</sup> ) = 4647,50
$P_{max, inv} (W) = 60,00$
$PF = 0,80$
$HSP = 4,6475$

Número de paneles
$E_{panel} (W) = 223,08$
N paneles = 0,828
N paneles elegidos = 1

Dimensionado del regulador
$V_{in} = 22,7$
$I_{SC \text{ PANEL}} = 3,48$
$I_{in} = 4,35$

Protecciones
$I_{in} = 4,872$

Inclinación elegida $\beta_{opt}$
Acimut $\phi$
Latitud $\delta$

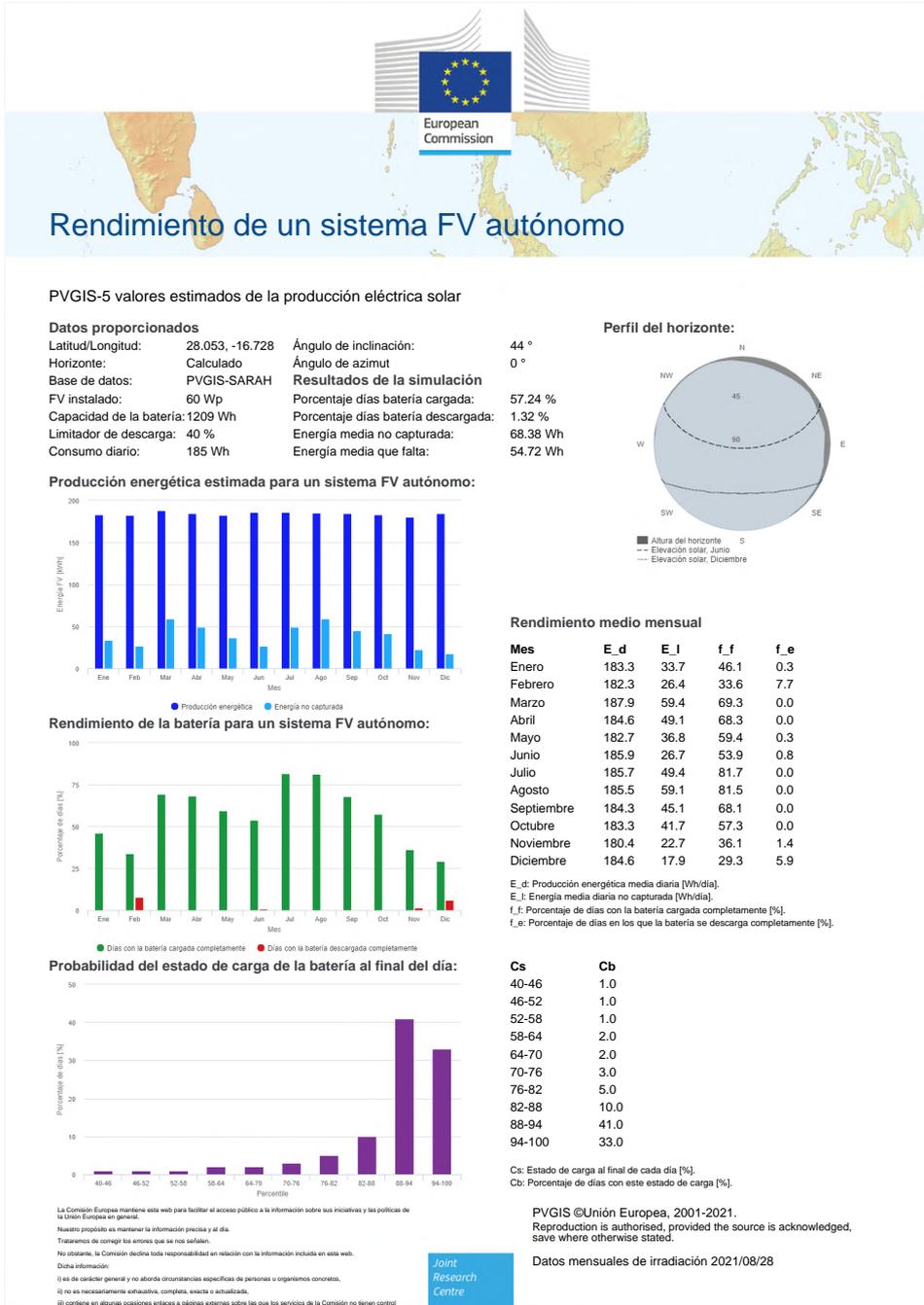
  

Consumo irradiación	
Enero	0,0621
Febrero	0,0391
Marzo	0,0353
Abril	0,0374
Mayo	0,0356
Junio	0,0325
Julio	0,0354
Agosto	0,0351
Septiembre	0,0363
Octubre	0,0399
Noviembre	0,0404
Diciembre	0,0401

Consumo irradiación											
	28°	29°	30°	32°	34°	36°	38°	40°	42°	44°	46°
Enero	0,0621	0,0618	0,0616	0,0611	0,0607	0,0604	0,0601	0,0599	0,0598	0,0597	0,0596
Febrero	0,0391	0,0379	0,0378	0,0376	0,0374	0,0372	0,0371	0,0370	0,0369	0,0369	0,0369
Marzo	0,0353	0,0353	0,0353	0,0353	0,0354	0,0355	0,0356	0,0357	0,0359	0,0362	0,0364
Abril	0,0395	0,0396	0,0396	0,0401	0,0404	0,0408	0,0412	0,0417	0,0422	0,0428	0,0435
Mayo	0,0374	0,0376	0,0378	0,0383	0,0389	0,0396	0,0401	0,0408	0,0416	0,0424	0,0433
Junio	0,0356	0,0358	0,0361	0,0367	0,0374	0,0381	0,0388	0,0396	0,0406	0,0415	0,0426
Julio	0,0325	0,0328	0,0330	0,0335	0,0341	0,0347	0,0354	0,0362	0,0370	0,0379	0,0388
Agosto	0,0354	0,0356	0,0358	0,0361	0,0365	0,0370	0,0375	0,0380	0,0386	0,0392	0,0399
Septiembre	0,0351	0,0351	0,0352	0,0353	0,0354	0,0356	0,0359	0,0361	0,0364	0,0367	0,0371
Octubre	0,0363	0,0362	0,0361	0,0359	0,0358	0,0357	0,0357	0,0356	0,0354	0,0357	0,0358
Noviembre	0,0399	0,0396	0,0394	0,0390	0,0386	0,0383	0,0380	0,0378	0,0376	0,0375	0,0373
Diciembre	0,0404	0,0401	0,0398	0,0392	0,0387	0,0383	0,0379	0,0376	0,0373	0,0370	0,0368

## G.2. Resultado de las simulaciones





## Apéndice H

### *Datasheet* del sistema de generación

En el presente anexo se muestran los *datasheets* del panel fotovoltaico **Xunzel PowerSolar 60Wp** y del cargador solar **EPSolar Tracer 3210AN**.

## H.1. Panel fotovoltaico Xunzel PowerSolar 60Wp



### SOLARPOWER™ Series. Paneles solares fotovoltaicos para carga de baterías

#### Especificaciones

Modelo	SOLARPOWER-5W-12V	SOLARPOWER-10W-12V	SOLARPOWER-15W-12V	SOLARPOWER-20W-12V	SOLARPOWER-20W-24V	SOLARPOWER-30W-12V	SOLARPOWER-30W-24V	SOLARPOWER-60W-12V	SOLARPOWER-80W-12V	SOLARPOWER-120W-12V	SOLARPOWER-185W-24V	SOLARPOWER-190W-12V
Potencia Máxima $P_{max}$ [W]	5	10	15	20	20	30	30	60	80	120	185	190
Tensión Nominal $V_{DC}$	12	12	12	12	24	12	24	12	12	12	24	12
Tensión Circuito Abierto $V_{OC}$ [V <sub>DC</sub> ]	22.00	21.20	22.00	21.96	43.20	22.00	43.20	22.70	23.00	22.70	43.95	23.60
Tensión MPP $V_{MPP}$ [V <sub>DC</sub> ]	18.00	17.60	18.70	18.30	36.00	18.70	36.00	18.30	18.30	18.30	36.50	19.58
Corriente Corto-circuito $I_{SC}$ [A <sub>DC</sub> ]	0.30	0.61	0.85	1.21	0.61	1.73	0.92	3.48	4.63	6.95	5.39	9.58
Corriente MPP $I_{MPP}$ [A <sub>DC</sub> ]	0.28	0.57	0.81	1.09	0.56	1.60	0.83	3.28	4.37	6.56	5.07	9.71
Tensión Máxima Sistema SCII $V_{DC}$	600	600	600	600	1000	600	1000	600	1000	1000	1000	1000
Dimensiones (mm)	240x180x17	340x240x25	345x240x25	420x340x25	420x340x24	604x340x25	505x430x25	674x604x30	768x676x35	1158x676x35	1230x992x35	1010x992x35
Peso Neto (kg)	0.60	1.00	1.40	1.60	2.00	2.20	2.80	5.10	6.70	9.30	13.80	11.50
Cables Solares Preinstalados	2 metros cable 2x 0.75 mm <sup>2</sup> (rojo+negro) (1 cable bifilar)		2 metros cable 2x 1.00 mm <sup>2</sup> (rojo + negro) (bifilar) SOLARPOWER-20-24 Y 30-25: 2x 0.75 mm <sup>2</sup> (rojo+negro) (1 cable bifilar)				4 metros cable solar 2 x 4 mm <sup>2</sup> (rojo + negro) (2-cables)					
Caja de Conexiones	IP65-67											
Rango de Temperatura Operacional	-40°C ~ +85°C (-40°F ~ +185°F)											
Temperatura de Operación Normalizada. NOCT	45 ± 2 °C											
Coefficiente de Tª para Potencia $P_{MPP}$	- 0.39%/K											
Coefficiente de Tª para Corriente $I_{DC}$	+ 0.05%/K											
Coefficiente de Tª para Voltaje $V_{DC}$	- 0.29%/K											
	No aplicar tensiones externas superiores al valor $V_{DC}$ al panel solar											
	Garantía de Potencia 10 años 90%, 20 años 80%   Marco de Aluminio Anodizado   Vidrio frontal 32mm (EN 12150) reforzado de alta resistencia, purificado bajo contenido en hierro, templado y laminado   Encapsulante EVA transparente de alta calidad   Láminas de alta calidad TPT / TPE / BBF   Inter-Conexión Células cobre alta pureza 99.97%											
	Fabricados de acuerdo con normas IEC61215 / IEC61730   Datos Eléctricos – Specifications at Standard Test Conditions (STC): 1000W/m <sup>2</sup> solar irradiance, AM 1.5, 25 °C cell temperature											
PLUS	SOLARPOWER 20W-24V y 30W-24V incluyen soporte integrado											

This document is for informational purposes only. Xunzel reserves the right to make changes and improvements without prior notice. ©XUNZEL. Xunzel is not responsible for any typographical or graphical errors. Specifications are subject to change without notice.

XU-2200709-MA

XUNZEL™ reserves the right to make changes and improvements without prior notice.

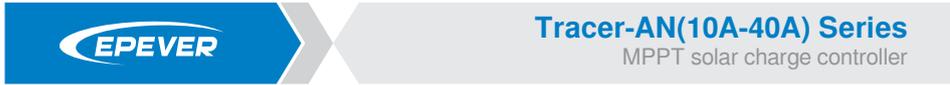
Industry Leading Technology for Off-Grid, Off-Shore and Backup Power Applications



© Copyright Xunzel. Information contained in this document is subject to change without notice.


 info@xunzel.com  
 www.xunzel.com

## H.2. Cargador MPPT 3210AN



### Overview

Tracer-AN(10A-40A)series charge controller adopts MPPT(Maximum Power Point Tracking)technology. In any situation, it can track the maximum power point (MPP) of the solar array and obtain the maximum solar energy rapidly and accurately. It can generate up to 30% more power compared with the PWM charge controller. This series can accept max.100V Voc and work with the solar panels which are designed for on-grid applications. The multifunction LCD display the system status vividly.

### Features

- MPPT tracking efficiency above 99.5%
- Maximum charge conversion efficiency as high as 98%
- Support lead-acid and lithium-ion batteries
- Multiple load work modes
- Charging power and current limitation function
- High-temperature charging power derating function
- Standard Modbus communication protocol with RS485 interface
- Real-time energy statistics function



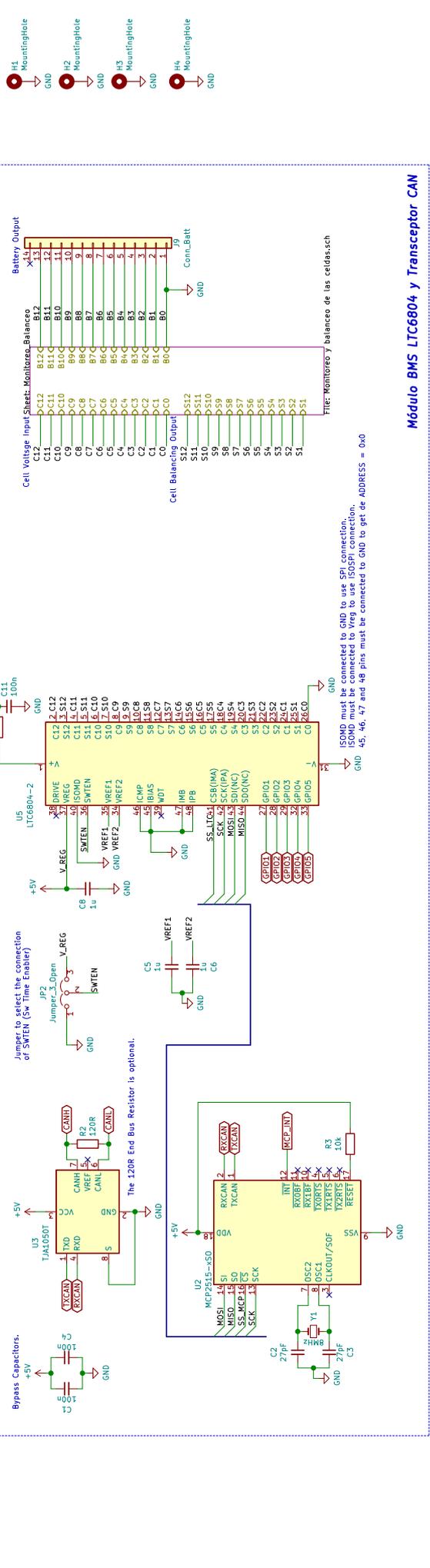


# Apéndice I

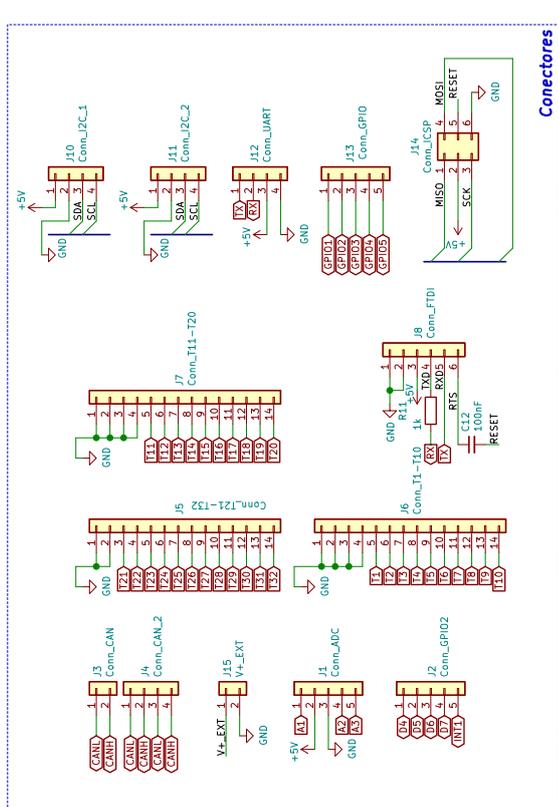
## Hojas de esquema del diseño electrónico

A continuación, se muestran los diferentes diseños electrónicos realizados. Se mostrarán los diferentes diseños realizados, siendo su orden el siguiente:

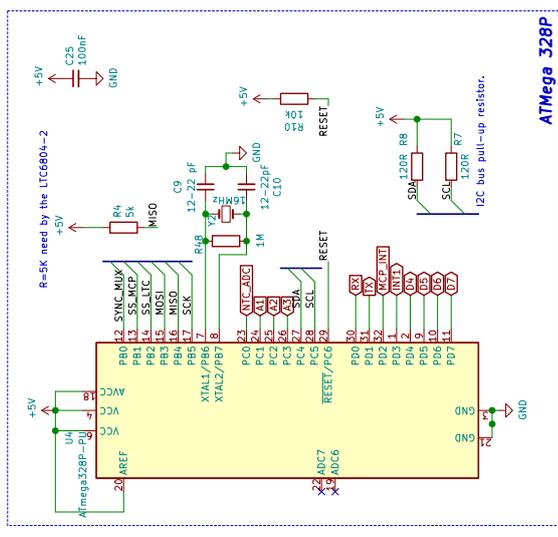
1. **Diseño del BMS:** Diseño realizado para el Sistema de Gestión de la Batería.
2. **Módulo CAN2Serial:** Diseño de la interfaz de comunicación de CAN a Serial.
3. **Módulo EPSolar-CAN:** Diseño de la interfaz de conversión de CAN a RS485 para el cargador solar.
4. **Programador ISP:** Diseño del programador empleado para el *flasheo* del *firmware* de los diferentes microcontroladores.



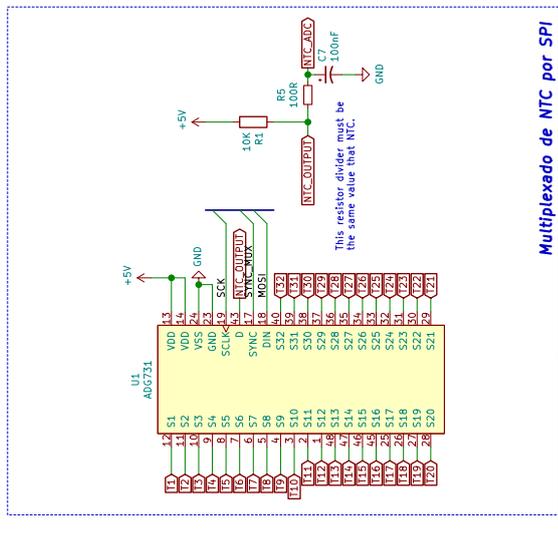
### Módulo BMS LTC6804 y Transceptor CAN



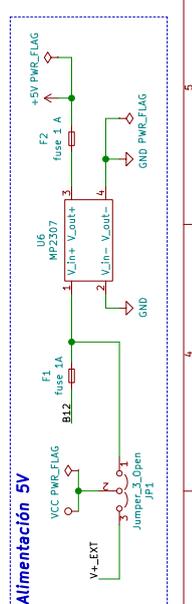
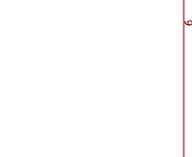
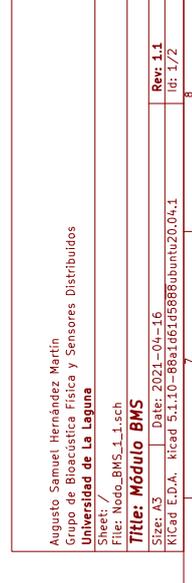
### Conectores



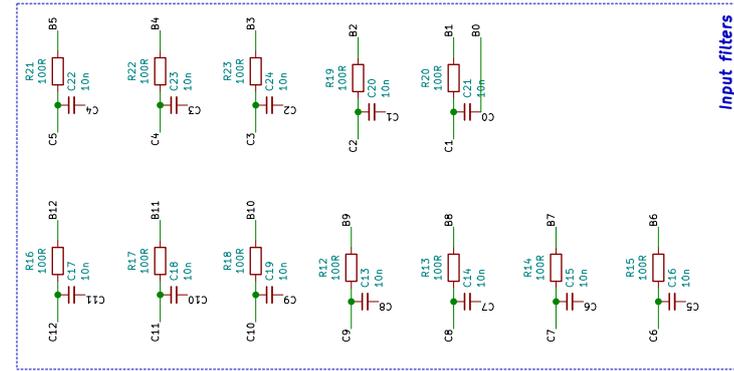
### ATMega 328P



### Multiplexado de NTC por SPI

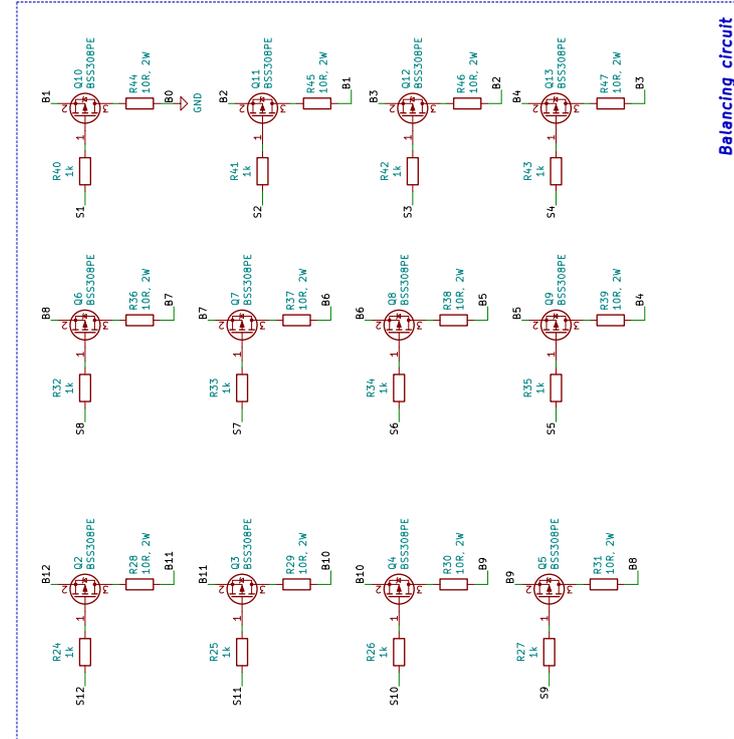


C12D C11  
C10D C10  
C9D C9  
C8D C8  
C7D C7  
C6D C6  
C5D C5  
C4D C4  
C3D C3  
C2D C2  
C1D C1  
C0D C0



Input filters

S12D S12  
S11D S11  
S10D S10  
S9D S9  
S8D S8  
S7D S7  
S6D S6  
S5D S5  
S4D S4  
S3D S3  
S2D S2  
S1D S1



Balancing circuit

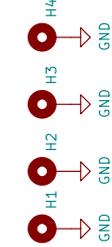
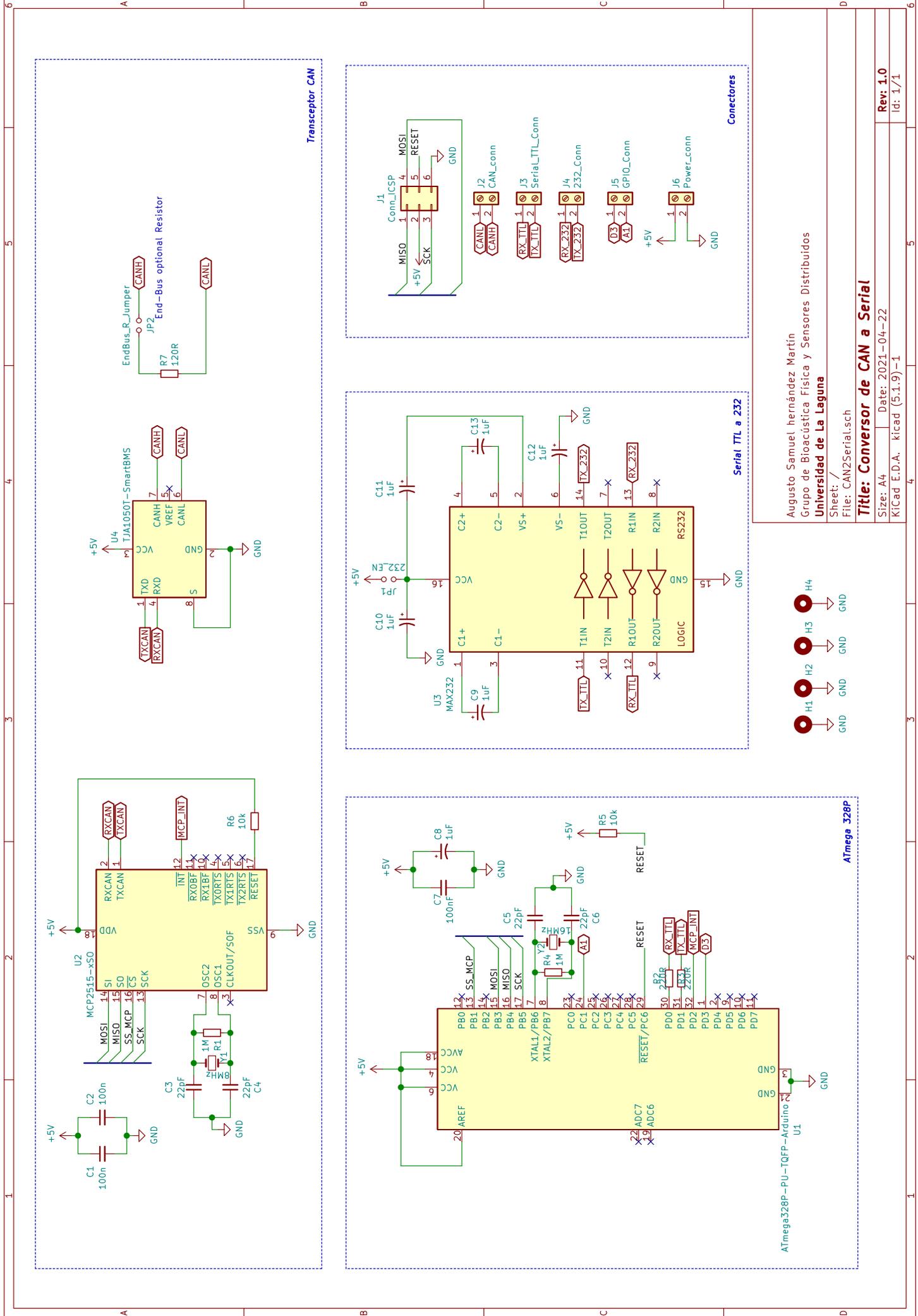
B12D B12  
B11D B11  
B10D B10  
B9D B9  
B8D B8  
B7D B7  
B6D B6  
B5D B5  
B4D B4  
B3D B3  
B2D B2  
B1D B1  
B0D B0

The battery cells should be connected half to the upper inputs and half to the lower inputs.  
The upper battery cell must always be connected to the input of its block and also to the B12 input to supply V+.  
The 100 Ohm resistor of input C12 and C6 must NEVER be unsoldered.  
In unused cells, the 10 nF capacitor must be replaced by a jumper and the 100 ohm resistor must be unsoldered.  
The transistors and balancing circuit may not be soldered on unused cells.

Augusto Samuel Hernández Martín  
Grupo de Bioacústica Física y Sensores Distribuidos  
Universidad de La Laguna  
Sheet: /Monitoreo\_Balanceo/  
File: Monitoreo y balanceo de las celdas.sch

Title: Monitoreo y Balanceo de Celdas  
Size: A4  
Date: 2021-01-28  
K:\Cad E.D.A. - kicad 5.1.10-88a1d61d5888buntku20.04.1

Rev: 1.0  
Id: 2/2

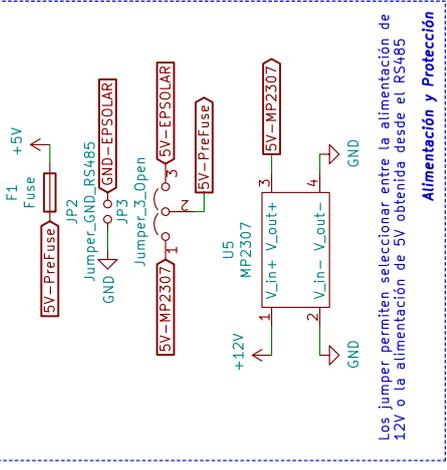
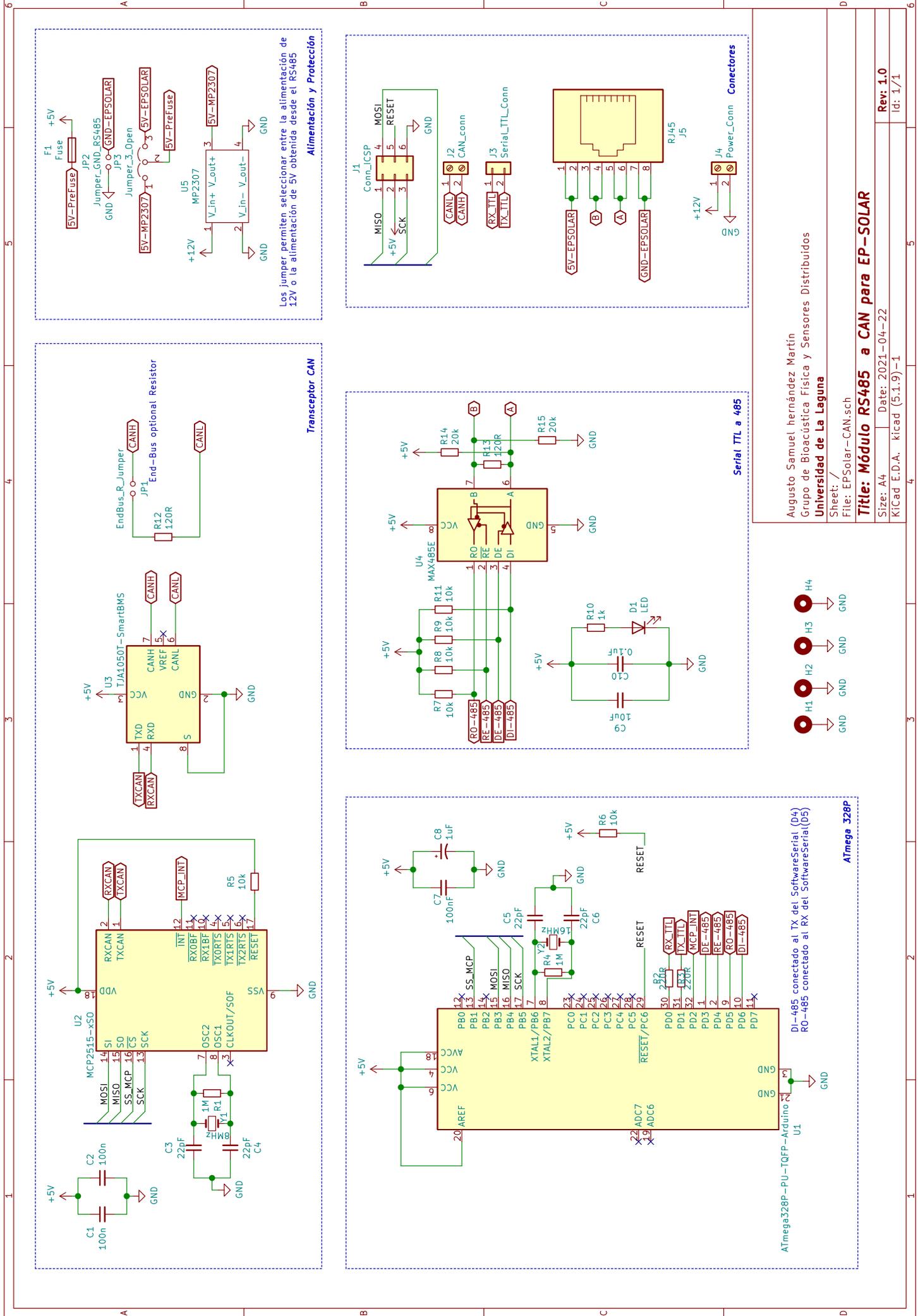


Augusto Samuel Hernández Martín  
 Grupo de Bioacústica Física y Sensores Distribuidos  
**Universidad de La Laguna**  
 Sheet: /  
 File: CAN2Serial.sch

**Title: Conversor de CAN a Serial**

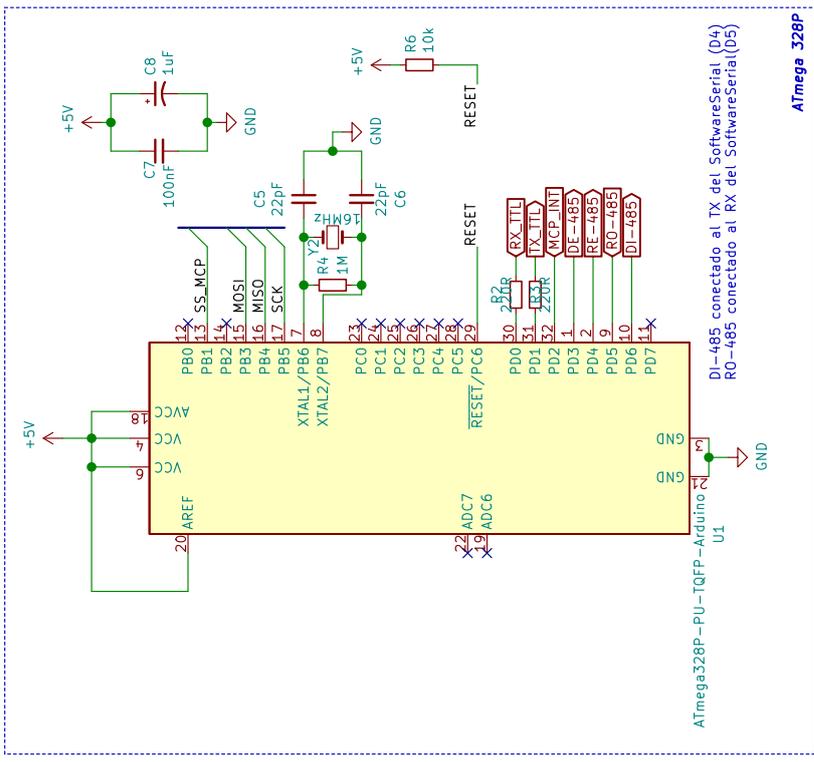
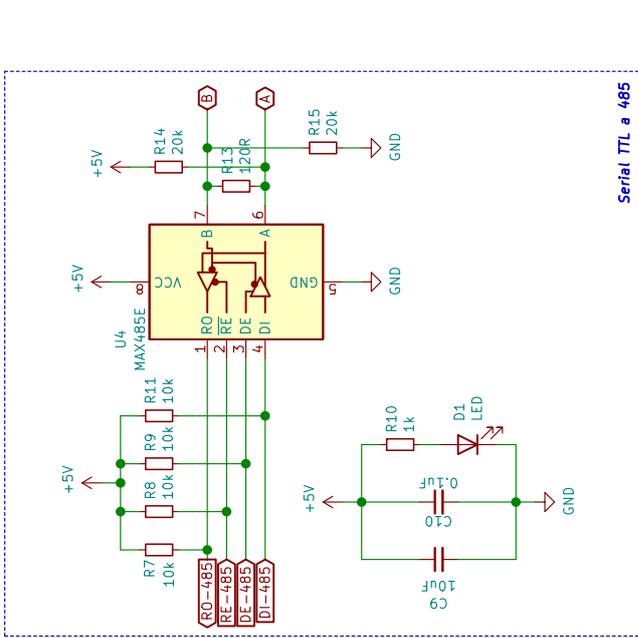
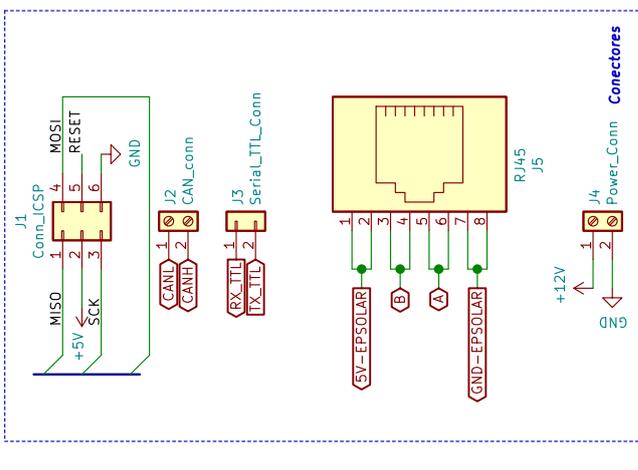
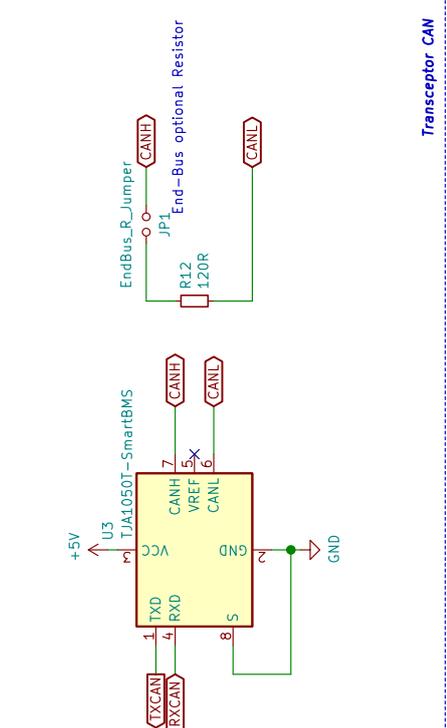
Size: A4 | Date: 2021-04-22  
 KiCad E.D.A. kicad (5:1.9)-1

Rev: 1.0  
 Id: 1/1



Los Jumper permiten seleccionar entre la alimentación de 12V o la alimentación de 5V obtenida desde el RS485

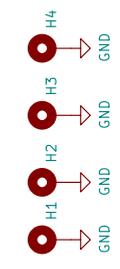
**Alimentación y Protección**



ATmega328P - PU - TQFP - Arduino U1

DL-485 conectado al TX del SoftwareSerial (D4)

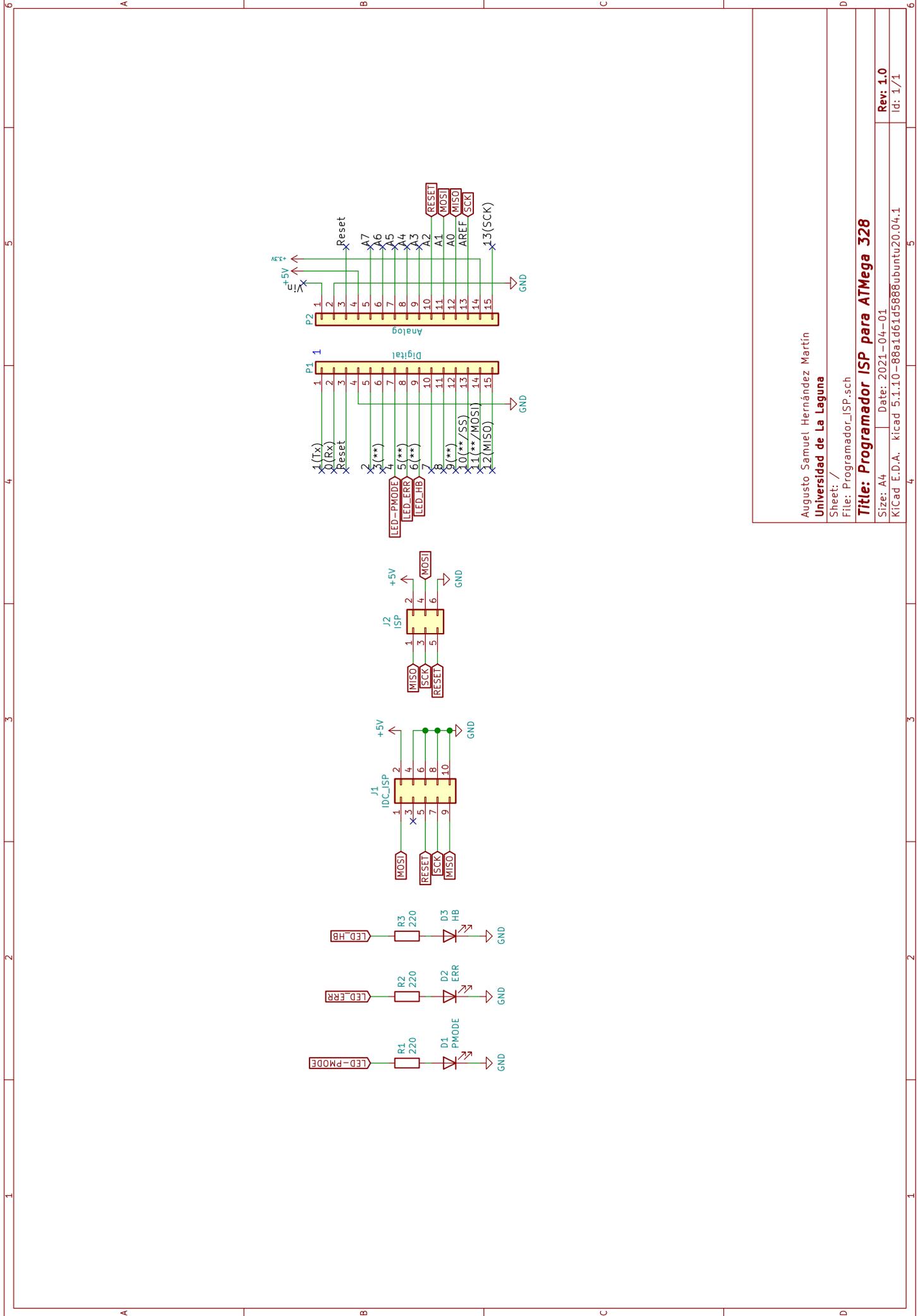
RO-485 conectado al RX del SoftwareSerial(D5)



Augusto Samuel Hernández Martín  
 Grupo de Bioacústica Física y Sensores Distribuidos  
**Universidad de La Laguna**

Sheet: /  
 File: EPSolar-CAN.isch  
**Title: Módulo RS485 a CAN para EP-SOLAR**  
 Size: A4 | Date: 2021-04-22  
 KiCad E.D.A. kicad (5:1.9)-1

Rev: 1.0  
 Id: 1/1



Augusto Samuel Hernández Martín  
**Universidad de La Laguna**

Sheet: /  
 File: Programador\_ISP.sch

**Title: Programador ISP para ATmega 328**

Size: A4 | Date: 2021-04-01  
 KiCad E.D.A. kicad 5.1.10-88a1d61d5888ubuntu20.04.1

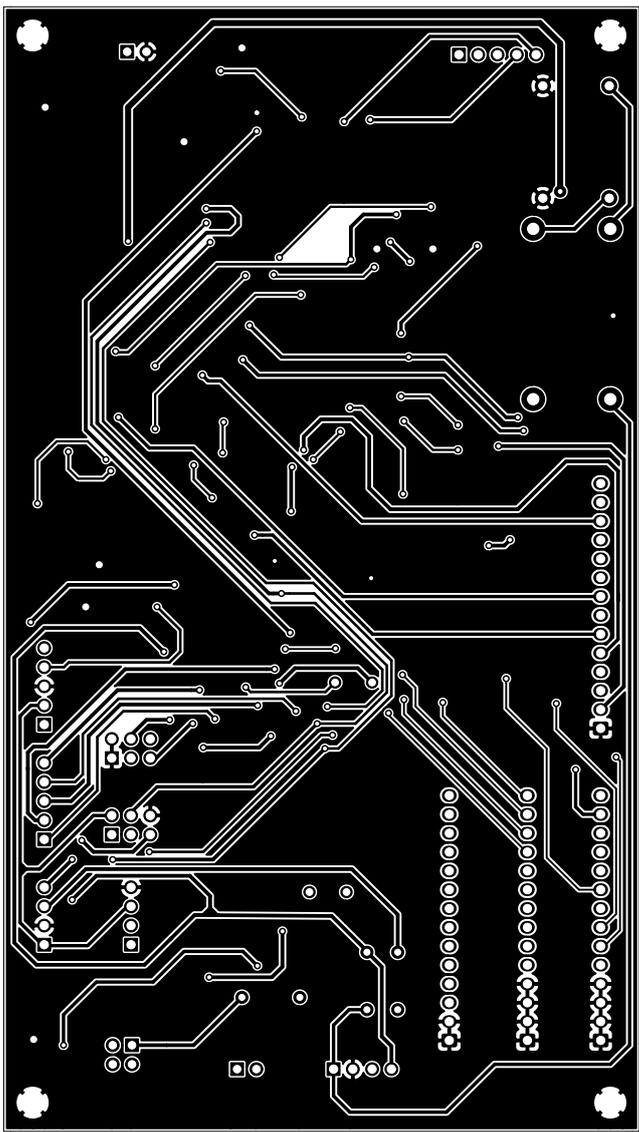
Rev: 1.0  
 Id: 1/1

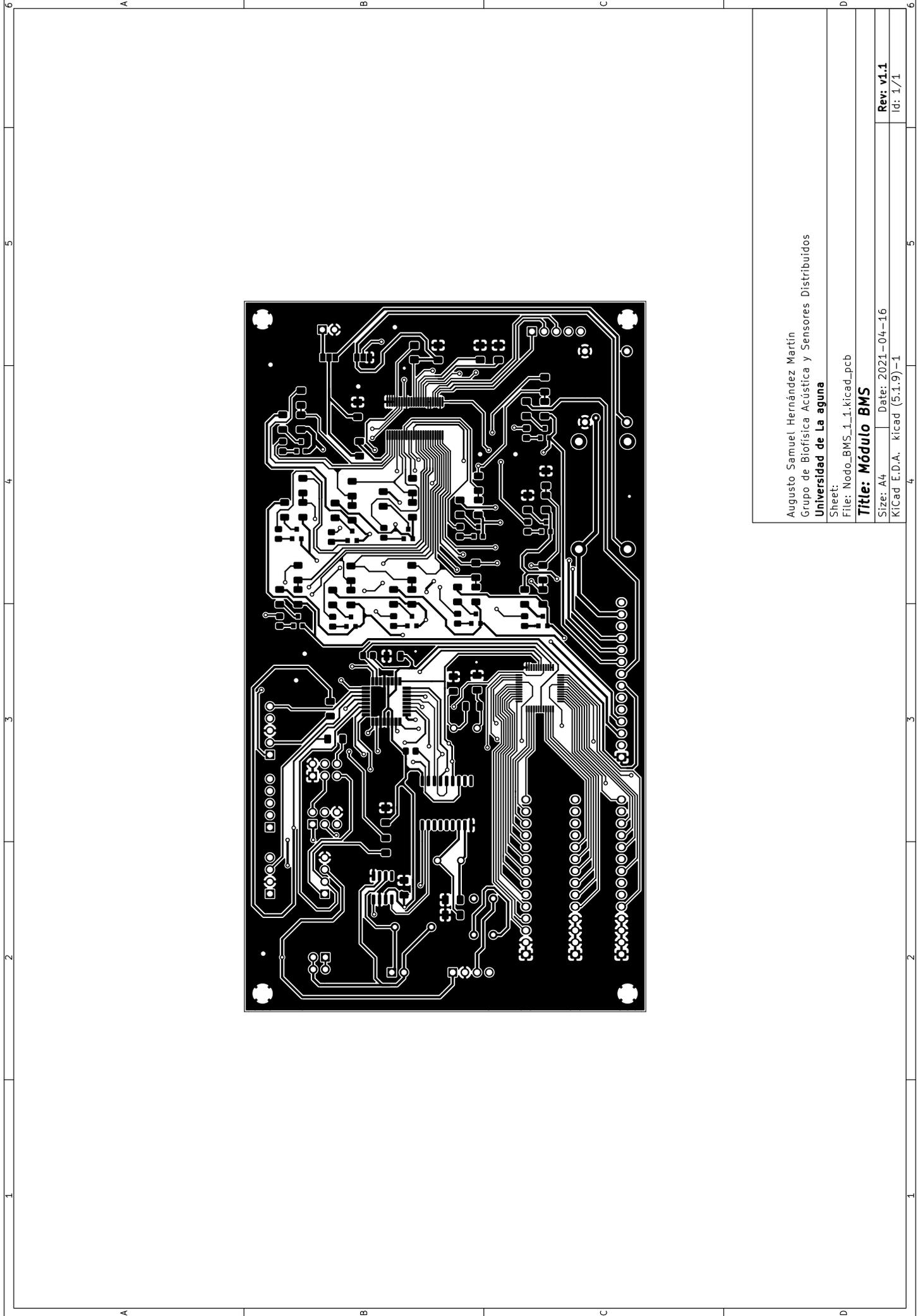
# Apéndice J

## Planos de diseño de placas de circuito impreso

A continuación, se muestran los fotolitos y planos requeridos para la fabricación de las placas de circuito impreso. Se incluyen las láminas conductoras de las capas *top* y *bottom*, además de la colocación de los elementos, la máscara de soldadura y la localización de los taladros. El orden de los planos presentados en este anexo es el siguiente:

1. **Diseño del BMS:** PCB realizada para el Sistema de Gestión de la Batería.
2. **Módulo CAN2Serial:** PCB realizada para la interfaz de comunicación de CAN a Serial.
3. **Módulo EPSolar-CAN:** PCB realizada para la interfaz de conversión de CAN a RS485 para el cargador solar.
4. **Programador ISP:** PCB realizada para el programador empleado para el *flasheo* del *firmware* de los diferentes microcontroladores.

6	A	B	C	D
5				
4	<p>Augusto Samuel Hernández Martín          Grupo de Biofísica Acústica y Sensores Distribuidos  <b>Universidad de La aguna</b></p> <p>Sheet:          File: Nodo_BMS_1_1.kicad_pcb</p> <p><b>Title: Módulo BMS</b></p> <p>Size: A4    Date: 2021-04-16          KiCad E.D.A.    kicad (5.1.9)-1</p>			
3	<p>Rev: v1.1          Id: 1/1</p>			
2	4			
1	5			
6	A	B	C	D



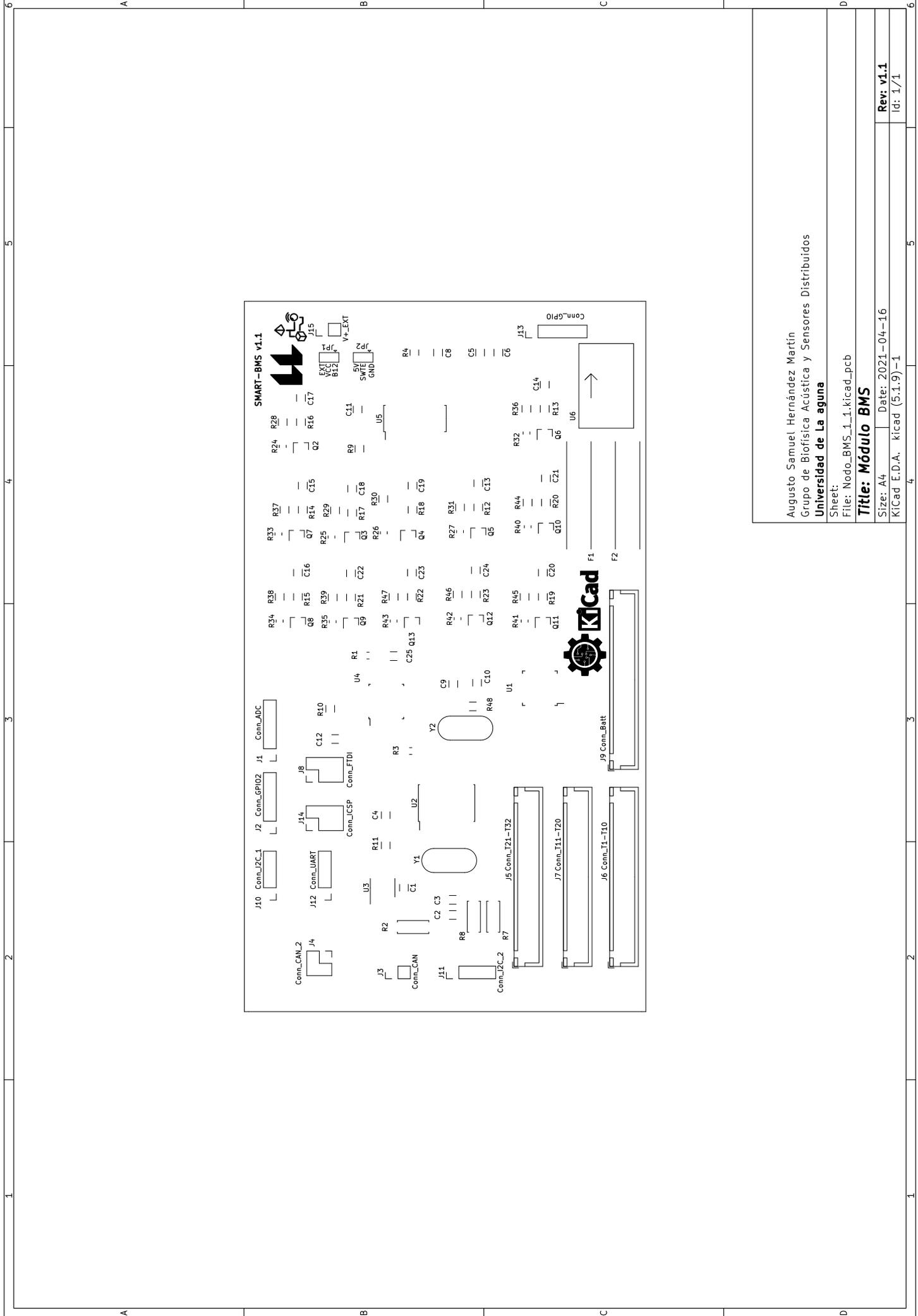
Augusto Samuel Hernández Martín  
Grupo de Física Acústica y Sensores Distribuidos  
**Universidad de La aguna**

Sheet:  
File: Nodo\_BMS\_1\_1.kicad\_pcb

**Title: Módulo BMS**

Size: A4 | Date: 2021-04-16  
KiCad E.D.A. kicad (5.1.9)-1

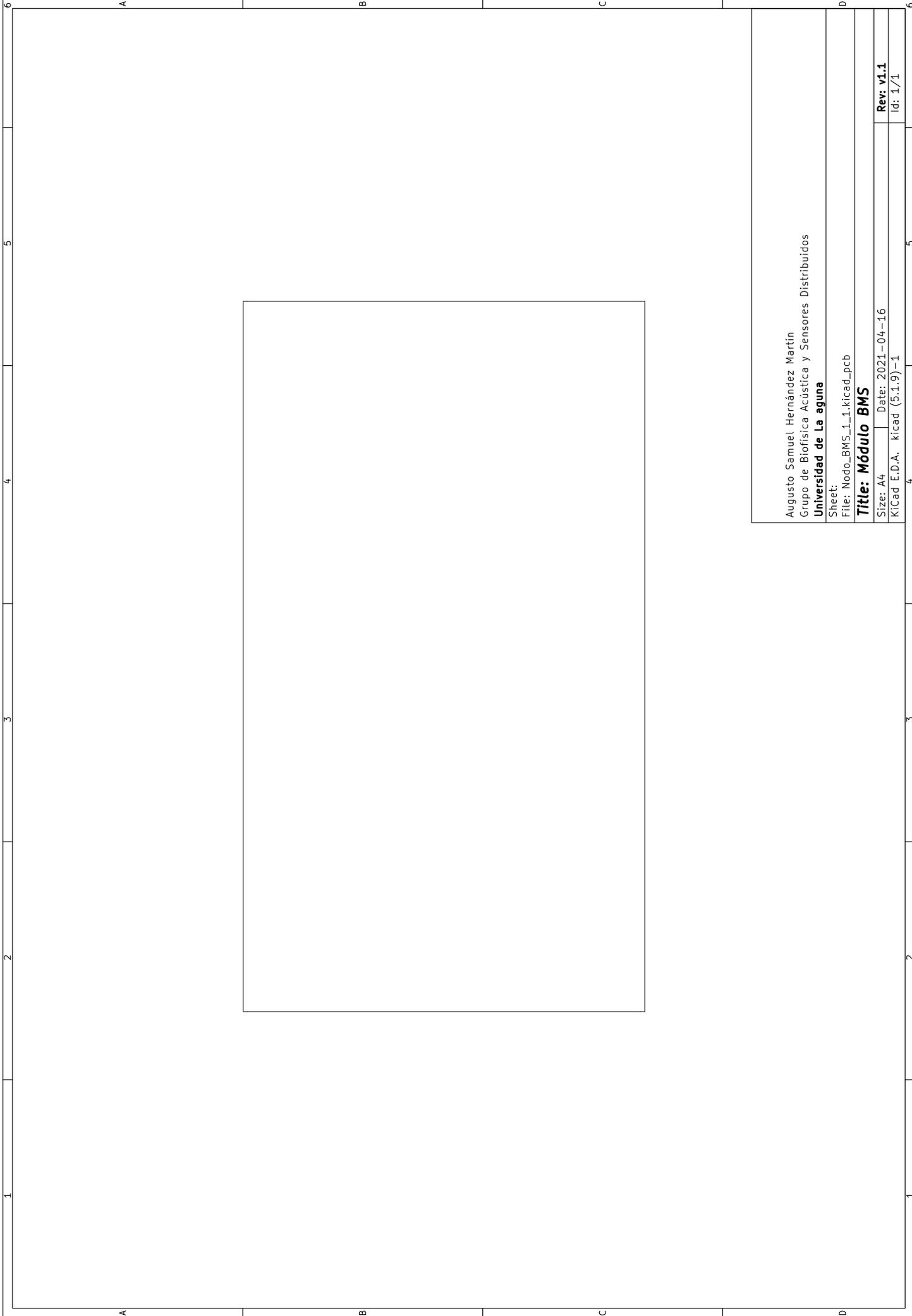
Rev: v1.1  
Id: 1/1



Augusto Samuel Hernández Martín  
 Grupo de Biofísica Acústica y Sensores Distribuidos  
**Universidad de La aguna**

Sheet:  
 File: Nodo\_BMS\_1.1.kicad\_pcb  
**Title: Módulo BMS**  
 Size: A4 | Date: 2021-04-16  
 KiCad E.D.A. kicad (5.1.9)-1

Rev: v1.1  
 Id: 1/1



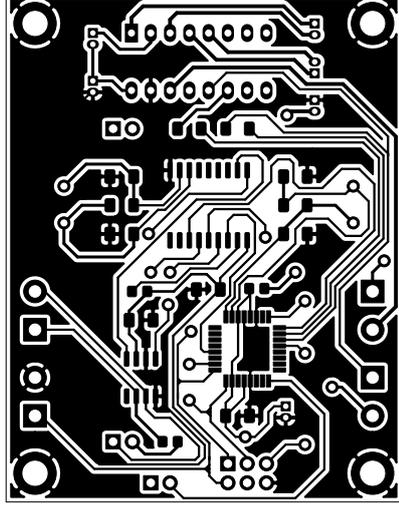
Augusto Samuel Hernández Martín  
Grupo de Biofísica Acústica y Sensores Distribuidos  
**Universidad de La aguna**

Sheet:  
File: Nodo\_BMS\_1\_1.kicad\_pcb

**Title: Módulo BMS**

Size: A4 | Date: 2021-04-16  
KiCad: E.D.A. kicad (5.1.9)-1

Rev: v1.1  
Id: 1/1



Augusto Samuel Hernández Martín  
**Universidad de La Laguna**

Sheet:

File: CAN2Serial.kicad\_pcb

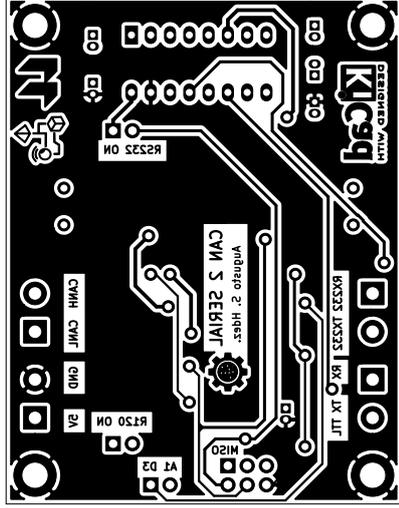
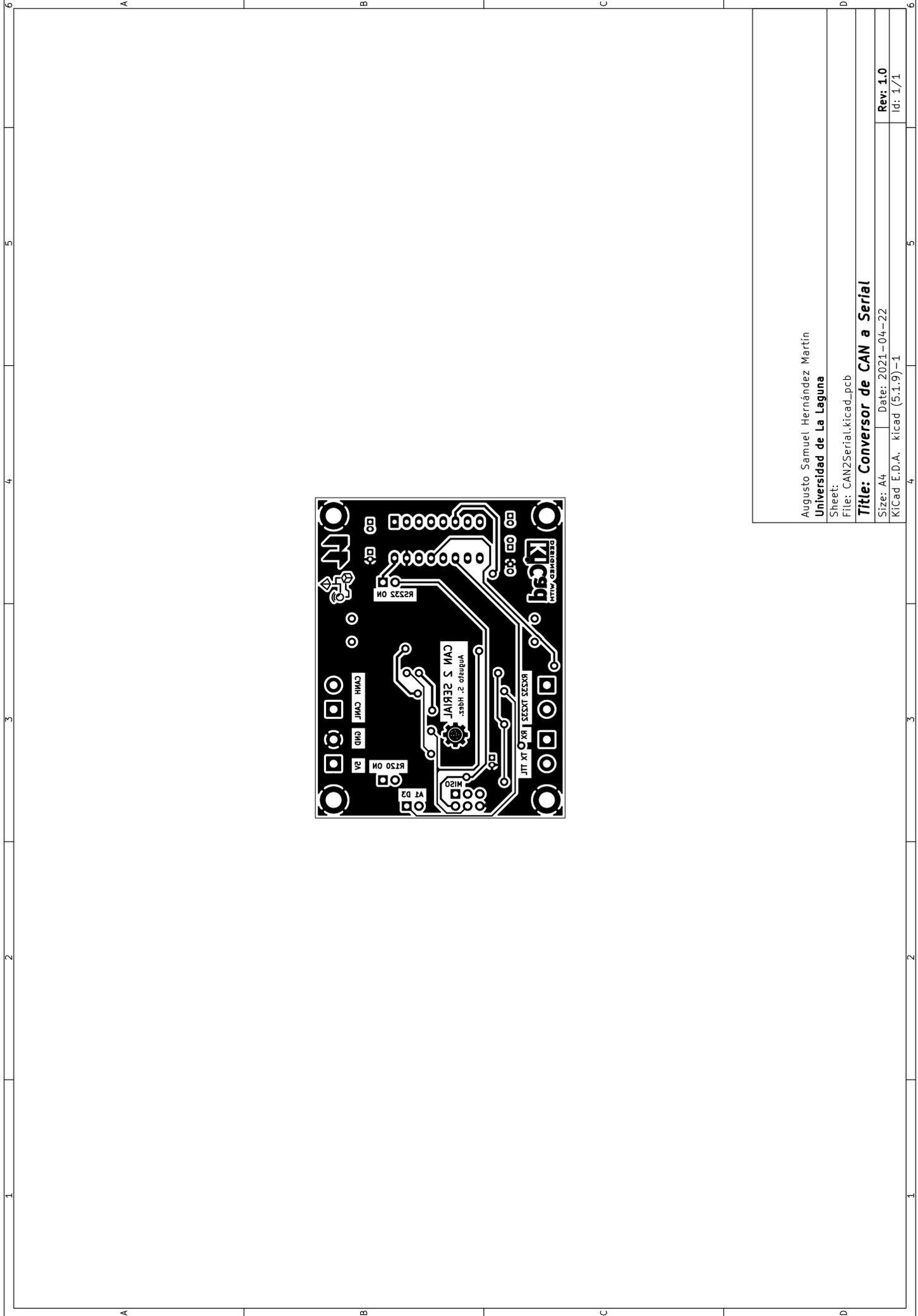
**Title: Conversor de CAN a Serial**

Size: A4 Date: 2021-04-22

KiCad E.D.A. kicad (5.1.9)-1

Rev: 1.0

Id: 1/1



Augusto Samuel Hernández Martín  
Universidad de La Laguna

Sheet:  
File: CAN2Serial.kicad\_pcb

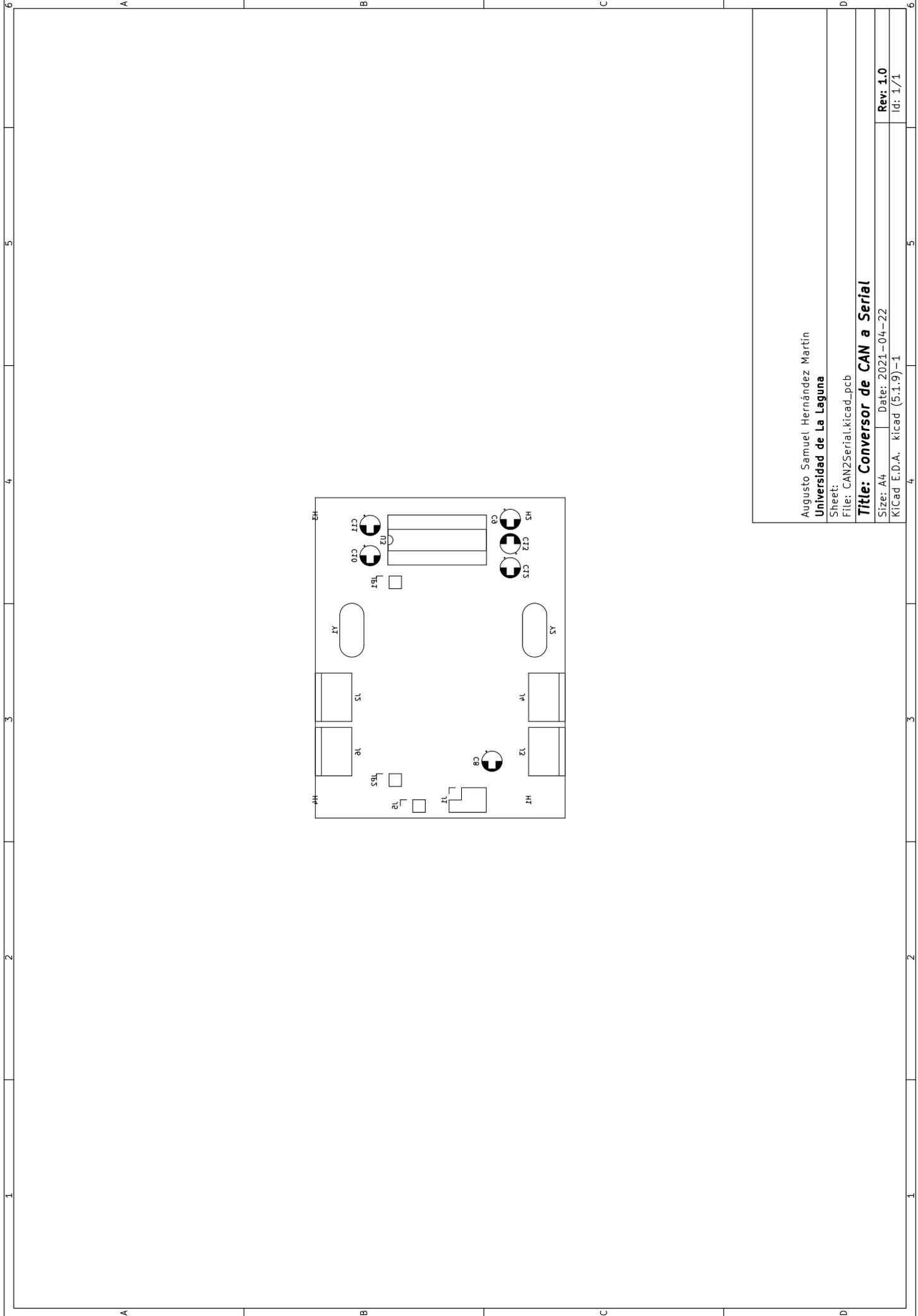
**Title: Conversor de CAN a Serial**

Size: A4 | Date: 2021-04-22

KiCad E.D.A. kicad (5.1.9)-1

Rev: 1.0

Id: 1/1



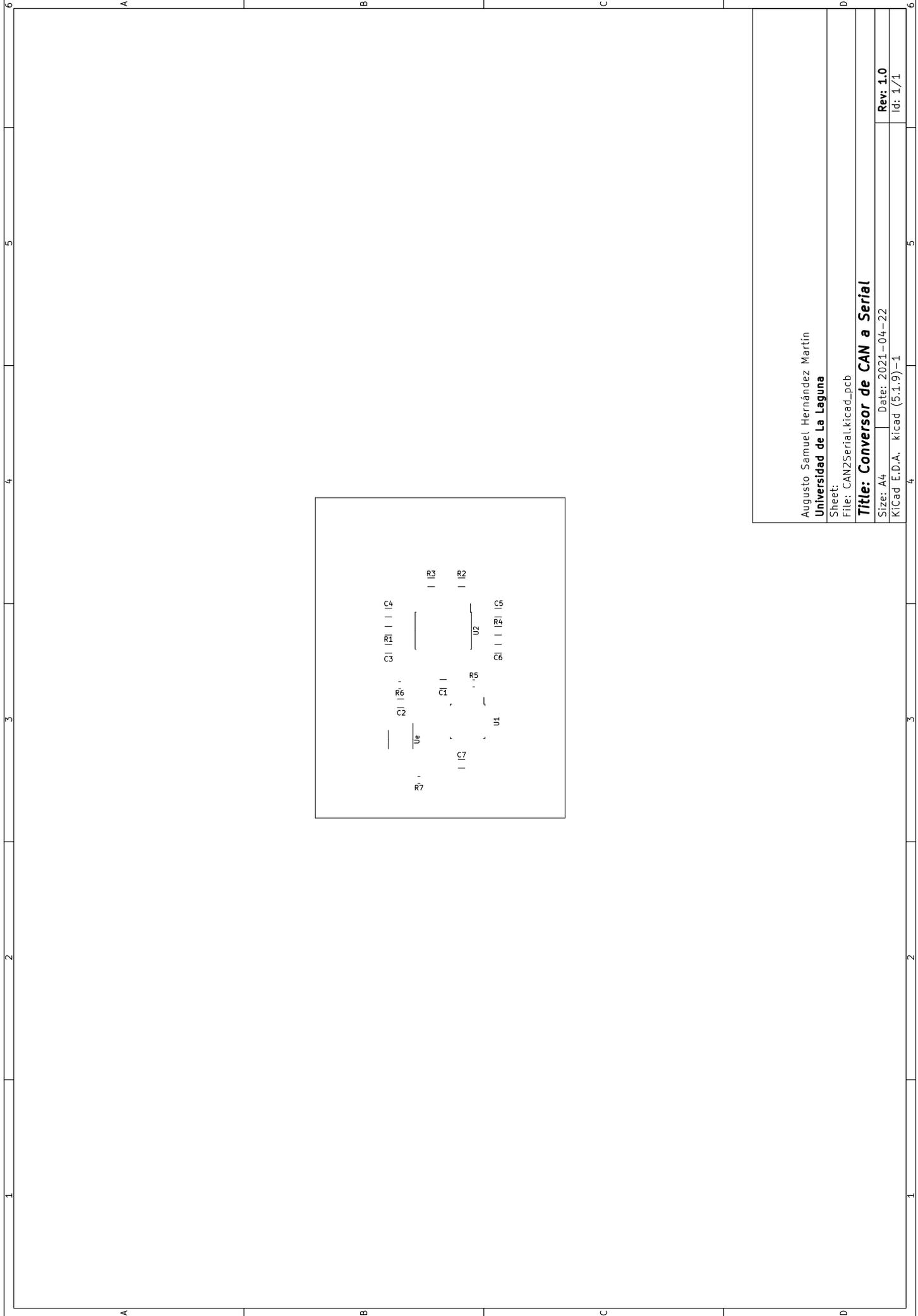
Augusto Samuel Hernández Martín  
**Universidad de La Laguna**

Sheet:  
 File: CAN2Serial.kicad\_pcb

**Title: Conversor de CAN a Serial**

Size: A4 | Date: 2021-04-22  
 KiCad E.D.A. kicad (5.1.9)-1

**Rev: 1.0**  
 Id: 1/1

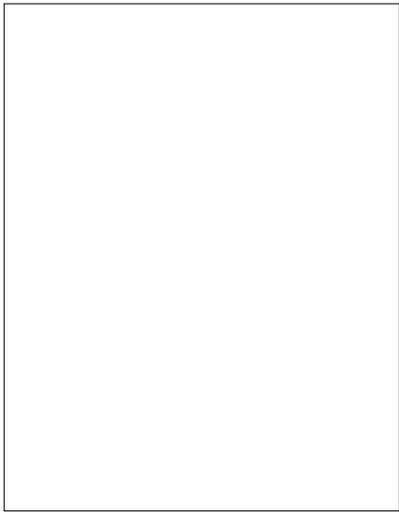


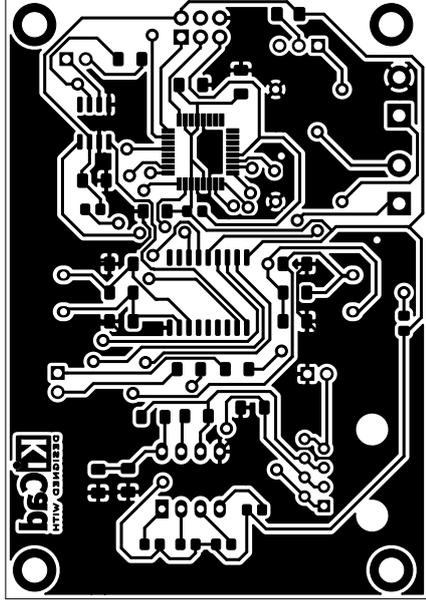
Augusto Samuel Hernández Martín  
**Universidad de La Laguna**

Sheet:  
 File: CAN2Serial.kicad\_pcb

**Title: Conversor de CAN a Serial**

Size: A4 | Date: 2021-04-22 | **Rev: 1.0**  
 KiCad E.D.A. kicad (5.1.9)-1 | Id: 1/1

6	A	B	C	D								
5												
4												
3	<p>Augusto Samuel Hernández Martín  <b>Universidad de La Laguna</b>  Sheet:  File: CAN2Serial.kicad_pcb  <b>Title: Conversor de CAN a Serial</b></p>											
2												
1	A	B	C	D								
6	A	B	C	D								
5	<table border="1"> <tr> <td data-bbox="1460 1355 1508 1377">Size: A4</td> <td data-bbox="1460 1377 1508 1400">Date: 2021-04-22</td> </tr> <tr> <td data-bbox="1460 1400 1508 1422">Kicad: E.D.A.</td> <td data-bbox="1460 1422 1508 1444">kicad (5.1.9)-1</td> </tr> <tr> <td colspan="2" data-bbox="1460 1444 1508 1467"><b>Rev: 1.0</b></td> </tr> <tr> <td colspan="2" data-bbox="1460 1467 1508 1489">Id: 1/1</td> </tr> </table>				Size: A4	Date: 2021-04-22	Kicad: E.D.A.	kicad (5.1.9)-1	<b>Rev: 1.0</b>		Id: 1/1	
Size: A4	Date: 2021-04-22											
Kicad: E.D.A.	kicad (5.1.9)-1											
<b>Rev: 1.0</b>												
Id: 1/1												
4	A	B	C	D								
3	<table border="1"> <tr> <td data-bbox="1460 1691 1508 1713">Size: A4</td> <td data-bbox="1460 1713 1508 1736">Date: 2021-04-22</td> </tr> <tr> <td data-bbox="1460 1736 1508 1758">Kicad: E.D.A.</td> <td data-bbox="1460 1758 1508 1780">kicad (5.1.9)-1</td> </tr> <tr> <td colspan="2" data-bbox="1460 1780 1508 1803"><b>Rev: 1.0</b></td> </tr> <tr> <td colspan="2" data-bbox="1460 1803 1508 1825">Id: 1/1</td> </tr> </table>				Size: A4	Date: 2021-04-22	Kicad: E.D.A.	kicad (5.1.9)-1	<b>Rev: 1.0</b>		Id: 1/1	
Size: A4	Date: 2021-04-22											
Kicad: E.D.A.	kicad (5.1.9)-1											
<b>Rev: 1.0</b>												
Id: 1/1												
2	A	B	C	D								
1	A	B	C	D								



Augusto Samuel Hernández Martín  
**Universidad de La Laguna**

Sheet:

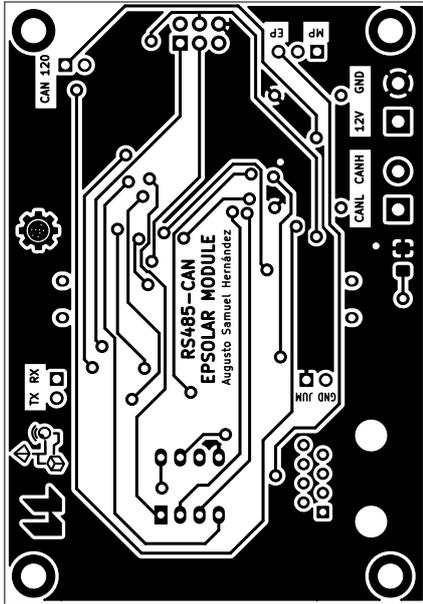
File: EPSolar-CAN.kicad\_pcb

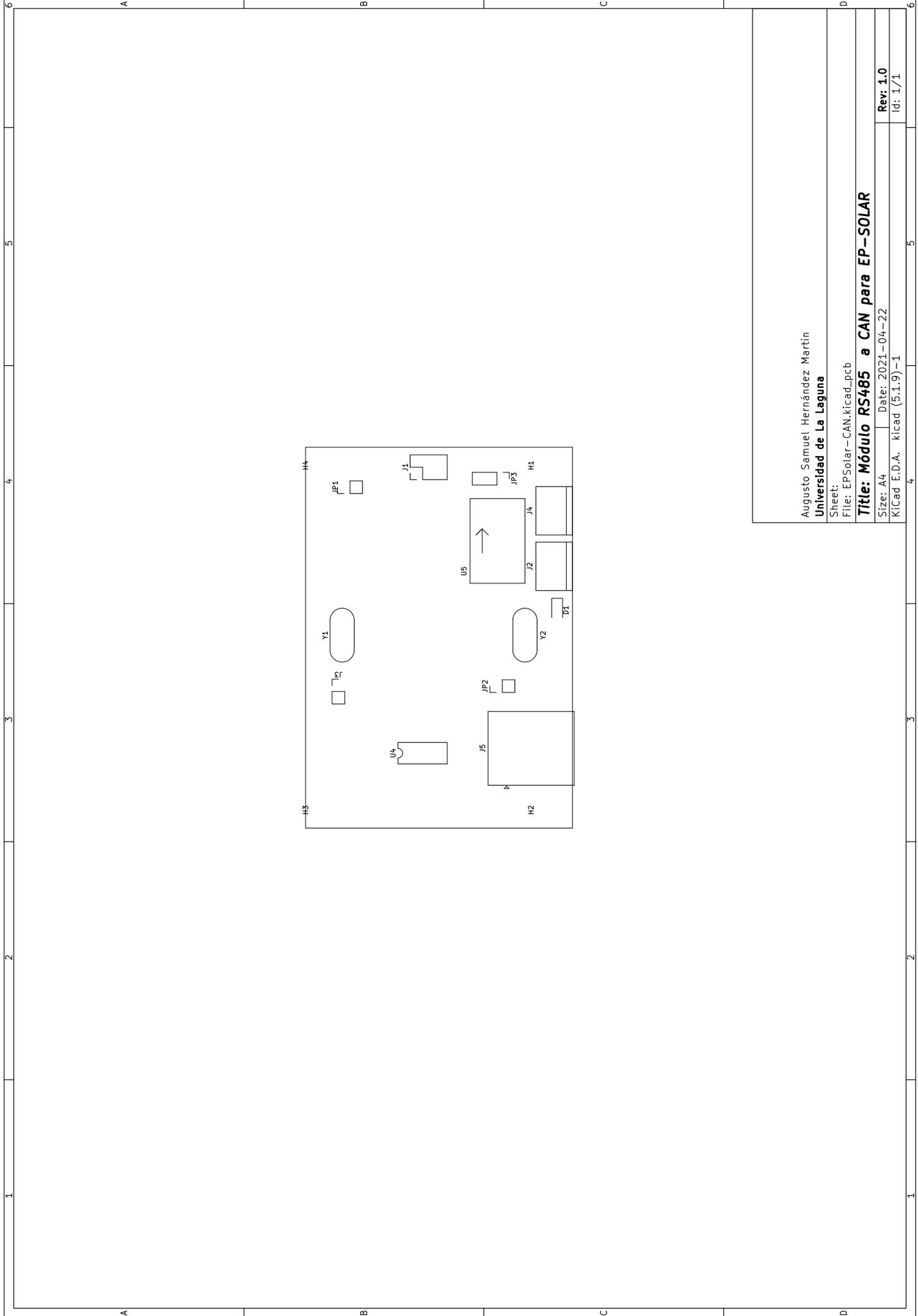
**Title: Módulo RS485 a CAN para EP-SOLAR**

Size: A4 Date: 2021-04-22 **Rev: 1.0**

KiCad: E.D.A. kicad (5.1.9)-1 Id: 1/1

1	2	3	4	5	6
A	A	B	B	C	C
D	D	<p>Augusto Samuel Hernández Martín  <b>Universidad de La Laguna</b></p> <p>Sheet:  File: EPSolar-CAN.kicad_pcb</p> <p><b>Title: Módulo RS485 a CAN para EP-SOLAR</b></p> <p>Size: A4   Date: 2021-04-22  KICad: E.D.A.   kicad (5.1.9)-1</p>			
1	2	3	4	5	6



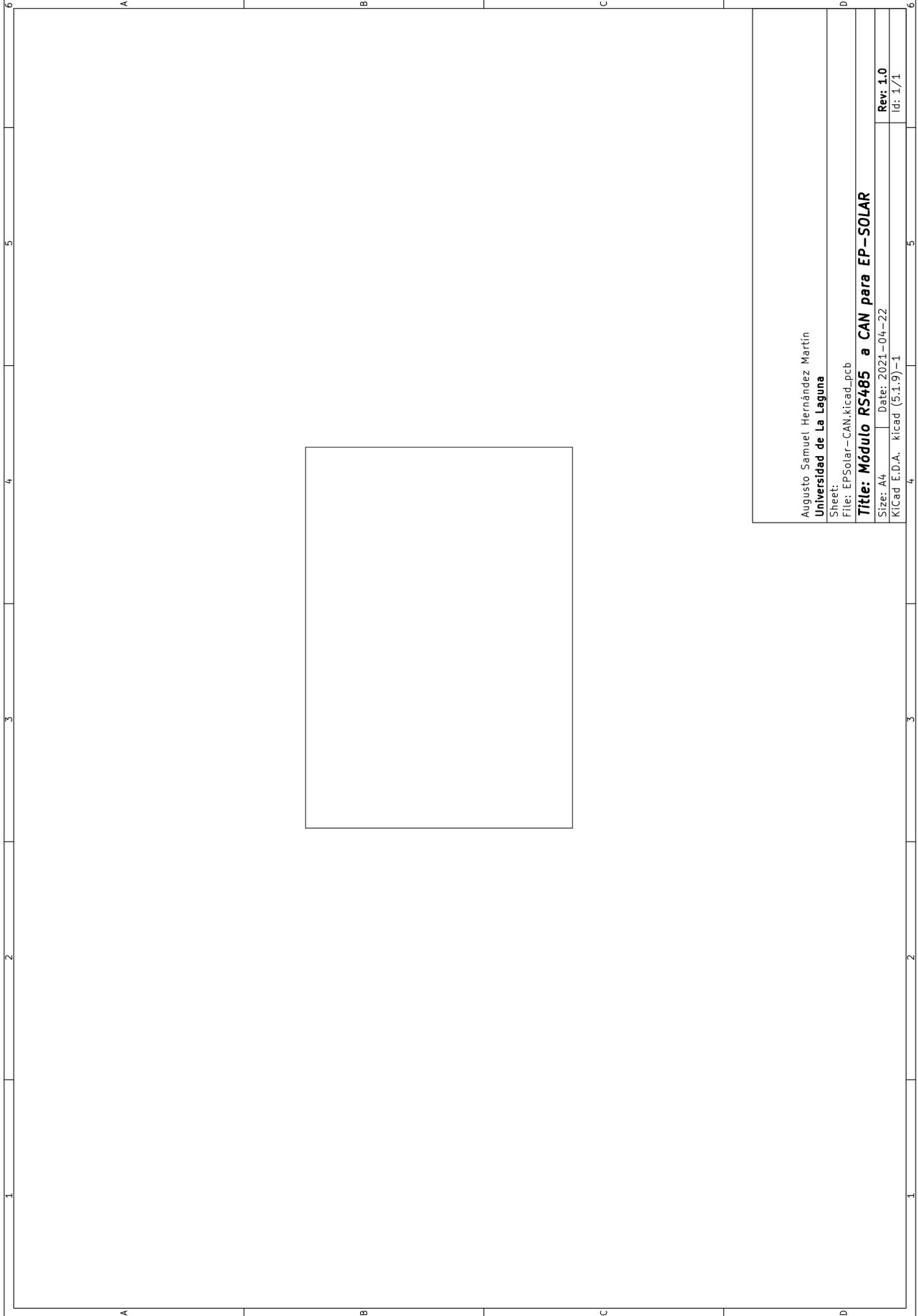


Augusto Samuel Hernández Martín  
**Universidad de La Laguna**

Sheet:  
 File: EPSolar-CAN.kicad\_pcb

**Title: Módulo RS485 a CAN para EP-SOLAR**

Size: A4 | Date: 2021-04-22 | **Rev: 1.0**  
 KiCad: E.D.A. kicad (5.1.9)-1 | Id: 1/1



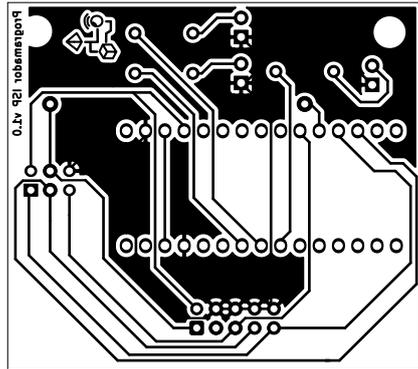
Augusto Samuel Hernández Martín  
**Universidad de La Laguna**

Sheet:  
File: EPSolar-CAN.kicad\_pcb

**Title: Módulo RS485 a CAN para EP-SOLAR**

Size: A4	Date: 2021-04-22
KiCad: E.D.A. kicad (5.1.9)-1	Rev: 1.0
	Id: 1/1

1	A								6
2									5
3									4
4									3
5									2
6									1
A									6
B									5
C									4
D									3



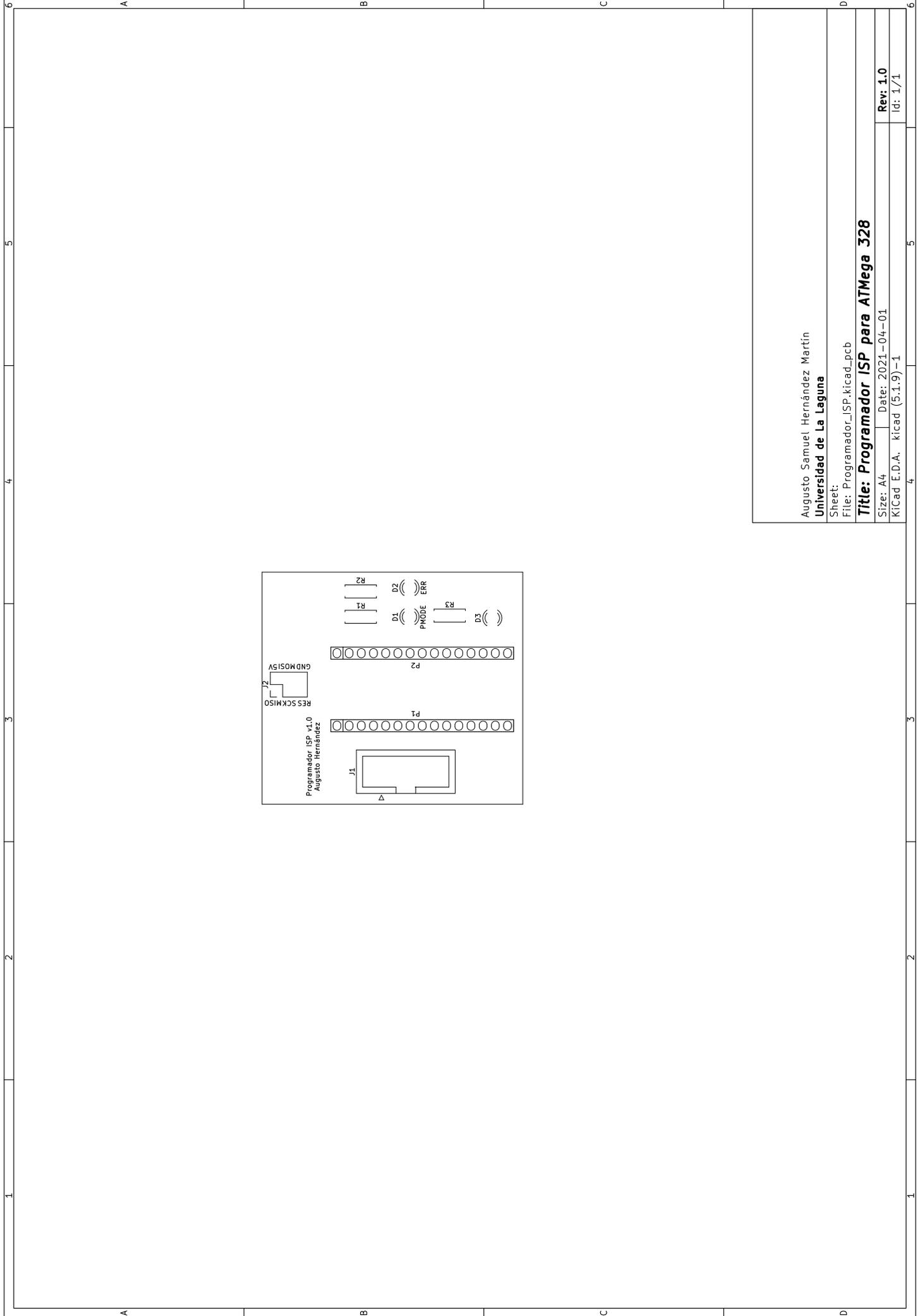
Augusto Samuel Hernández Martín  
**Universidad de La Laguna**

Sheet:  
 File: Programador\_ISP.kicad\_pcb

**Title: Programador ISP para ATmega 328**

Size: A4 | Date: 2021-04-01  
 KiCad E.D.A. kicad (5.1.9)-1

Rev: 1.0  
 Id: 1/1



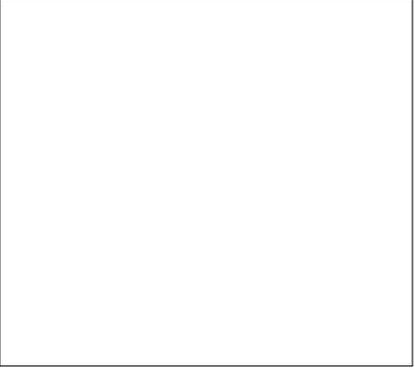
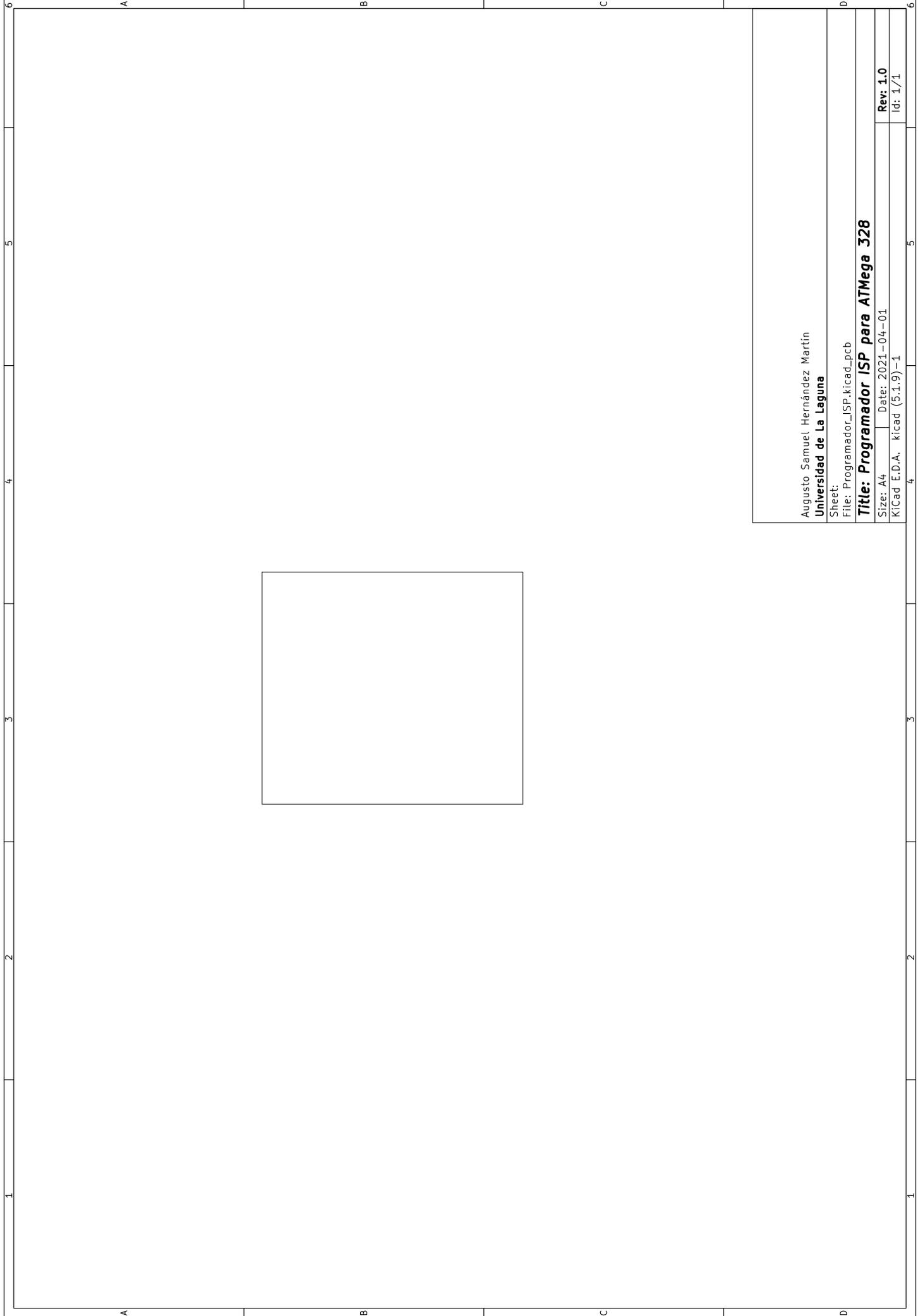
Augusto Samuel Hernández Martín  
**Universidad de La Laguna**

Sheet:  
 File: Programador\_ISP.kicad\_pcb

**Title: Programador ISP para ATmega 328**

Size: A4 | Date: 2021-04-01  
 KiCad E.D.A. kicad (5.1.9)-1

Rev: 1.0  
 Id: 1/1



Augusto Samuel Hernández Martín  
**Universidad de La Laguna**

Sheet:  
File: Programador\_ISP.kicad\_pcb

**Title: Programador ISP para ATmega 328**

Size: A4 | Date: 2021-04-01  
Kicad: E.D.A. kicad (5.1.9)-1

Rev: 1.0  
Id: 1/1



# Apéndice K

## Modelos 3D de los módulos desarrollados

En este anexo se recogen los modelos 3D de los diferentes módulos que se han diseñado. El orden de presentación de los modelos es el siguiente:

1. Modelos 3D del módulo de Gestión de Batería (BMS).
2. Modelos 3D del adaptador CAN2Serial.
3. Modelos 3D del adaptador EPSolar-CAN.
4. Modelos 3D del programador ISP.

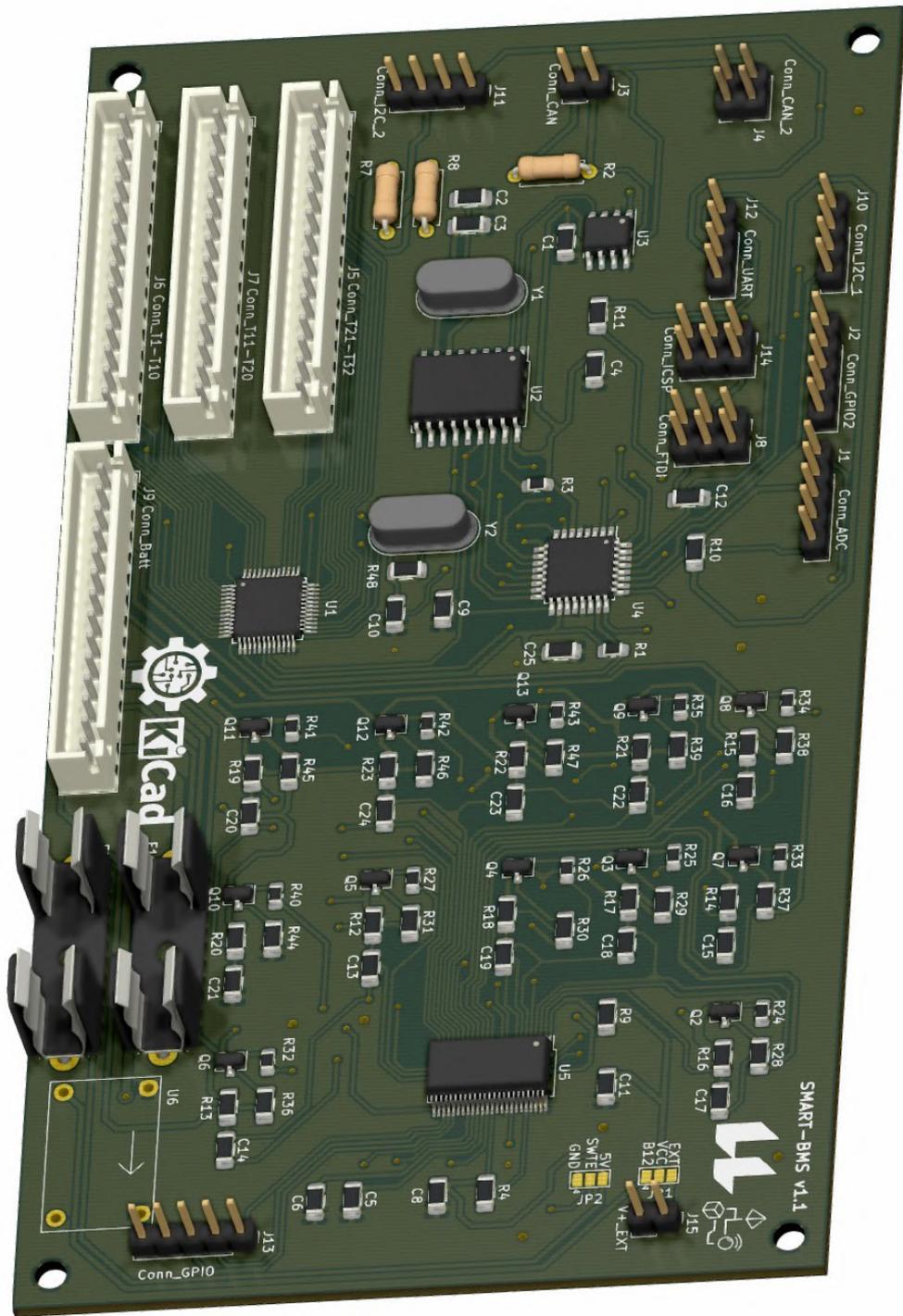


Figura K.1: Modelo 3D del BMS. Vista superior.

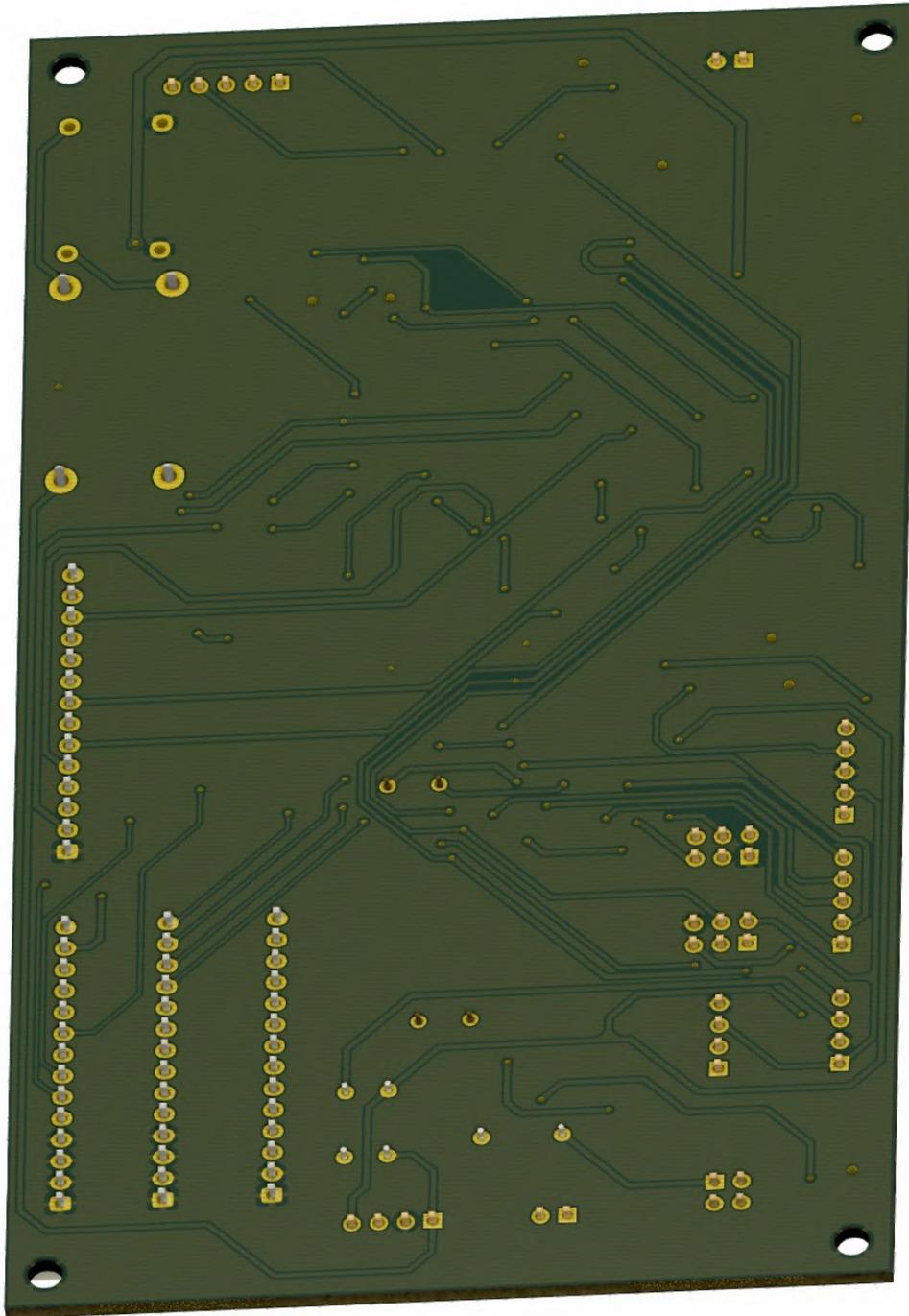


Figura K.2: Modelo 3D del BMS. Vista inferior.

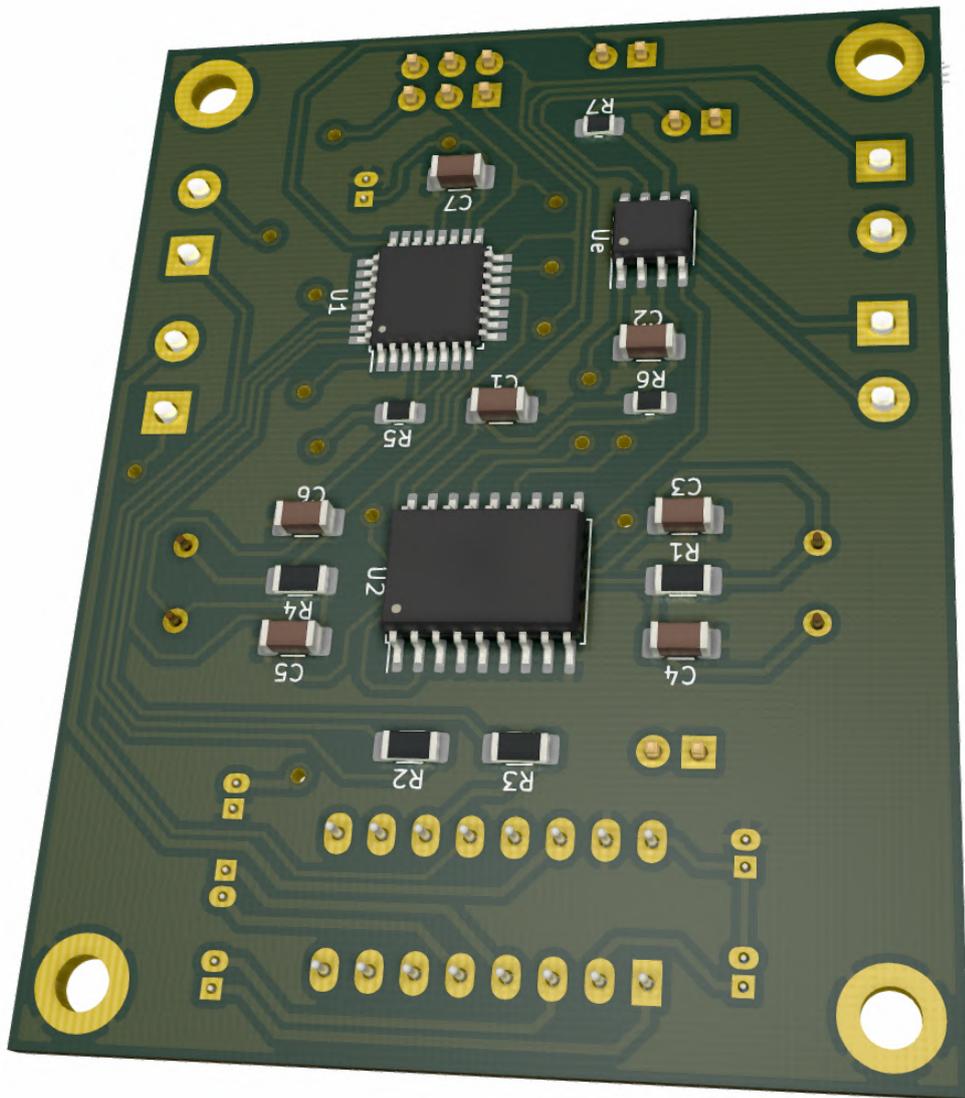


Figura K.3: Modelo 3D del CAN2Serial. Vista superior.

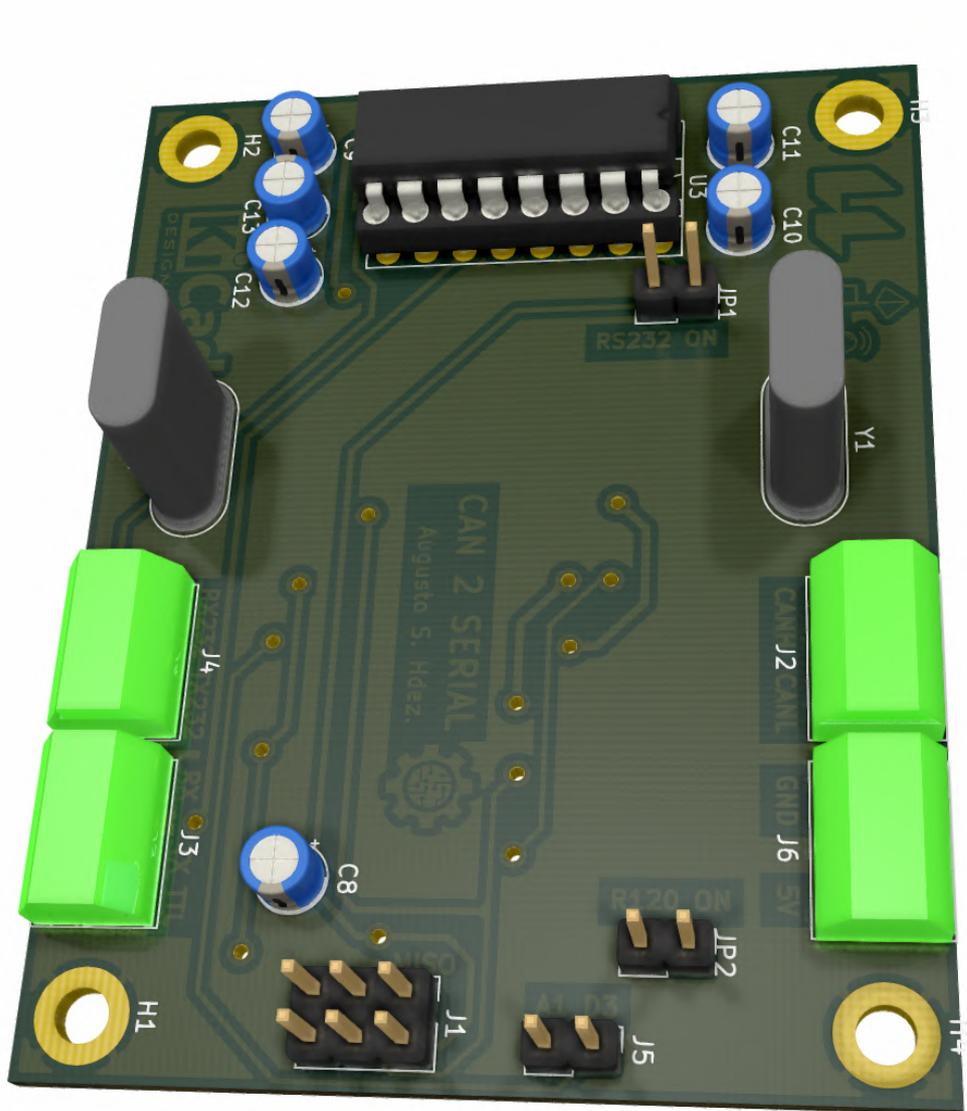


Figura K.4: Modelo 3D del CAN2Serial. Vista inferior.

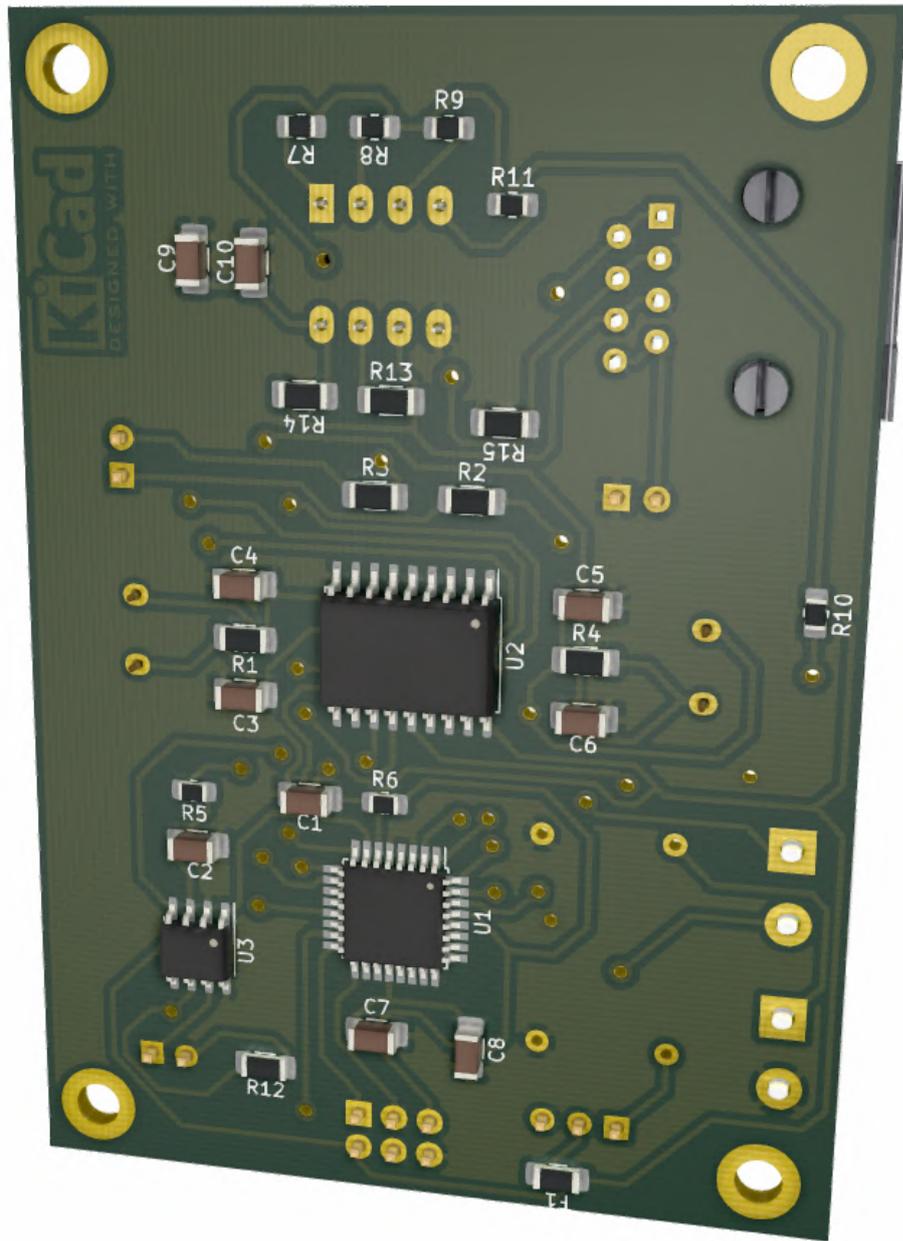


Figura K.5: Modelo 3D del EPSolar-CAN. Vista superior.



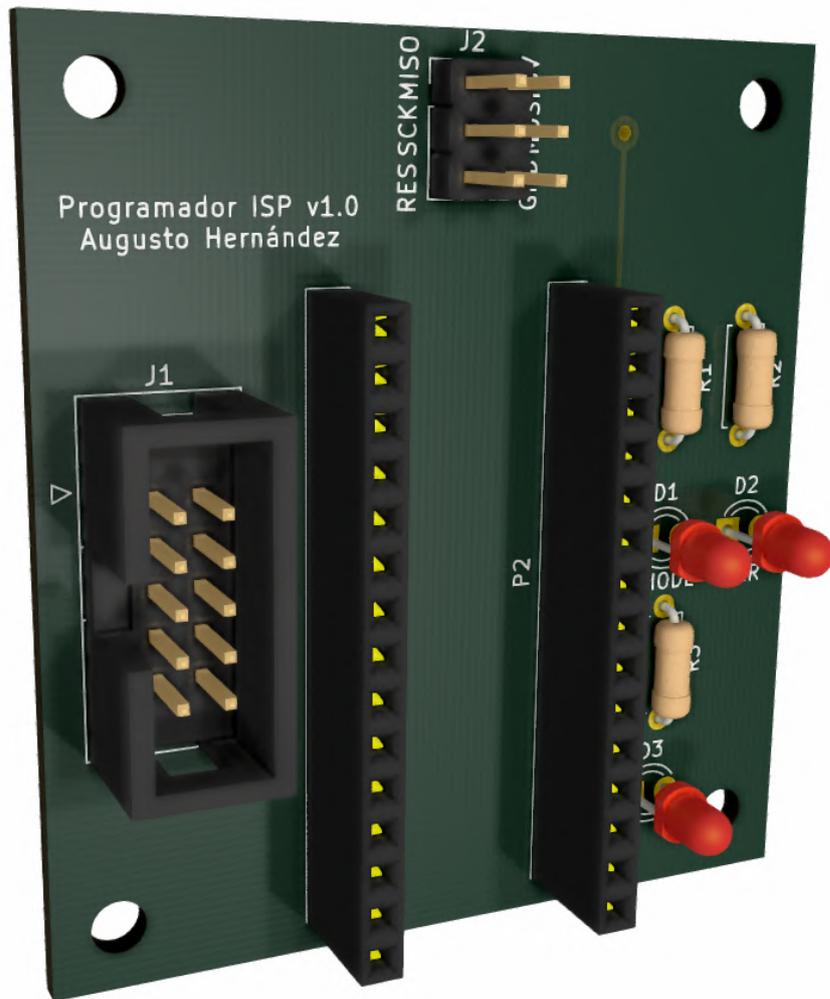


Figura K.7: Modelo 3D del Programador ISP. Vista superior.



Figura K.8: Modelo 3D del Programador ISP. Vista inferior.

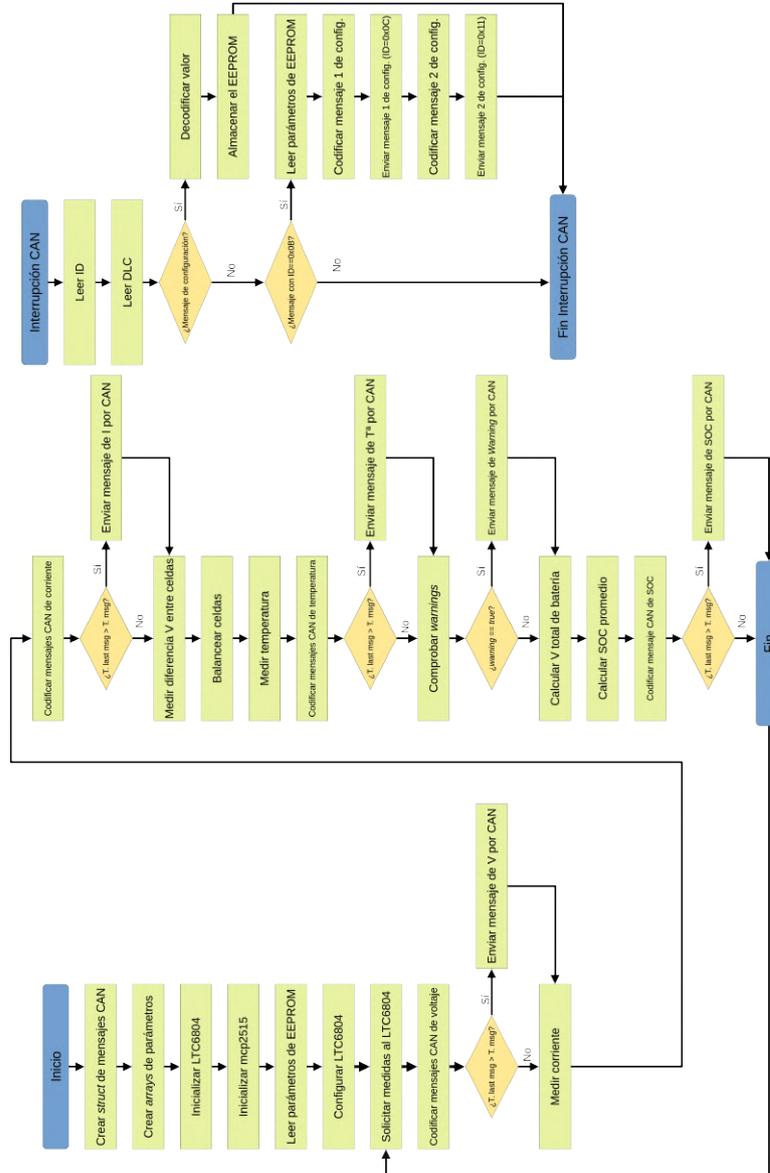


## **Apéndice L**

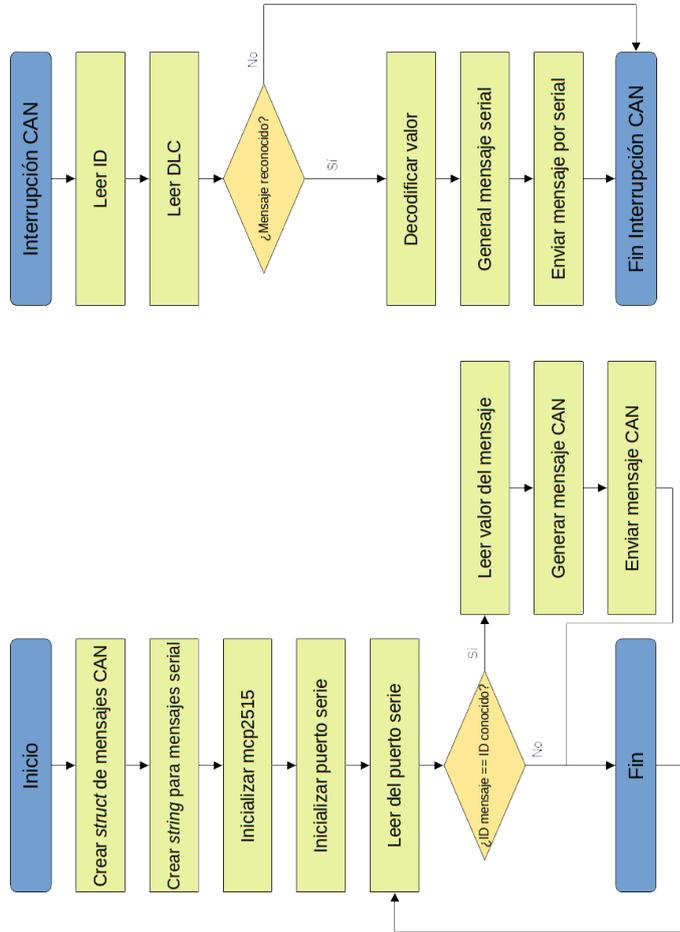
### **Diagramas de flujo del *firmware* desarrollado**

En el presente anexo se presentan los diagramas de flujo de funcionamiento del BMS y del módulo CAN2Serial.

## L.1. Diagrama de flujo del BMS



## L.2. Diagrama de flujo de CAN2Serial



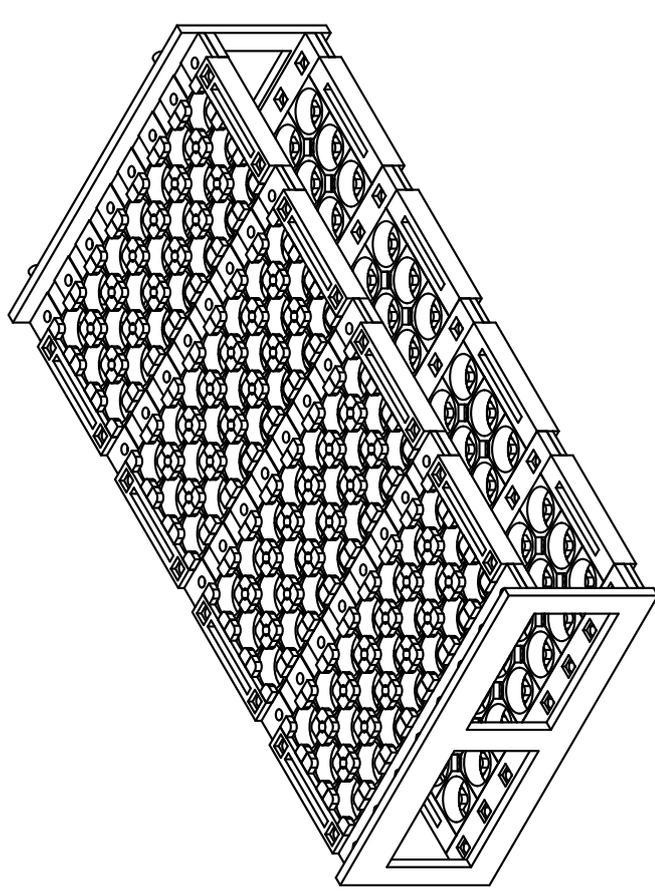
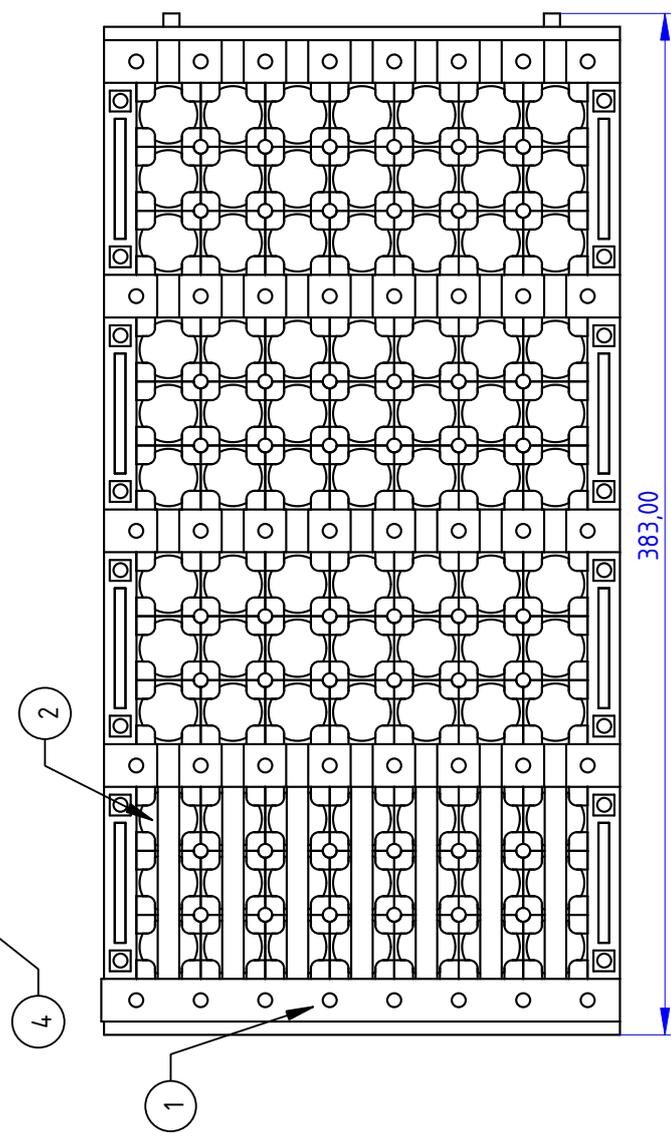
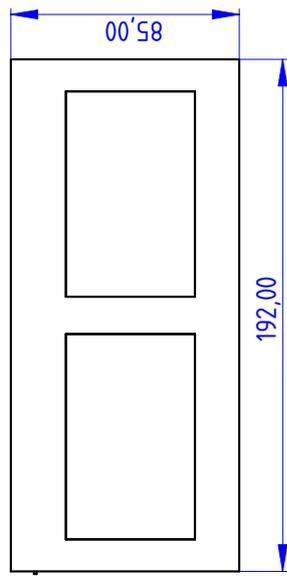
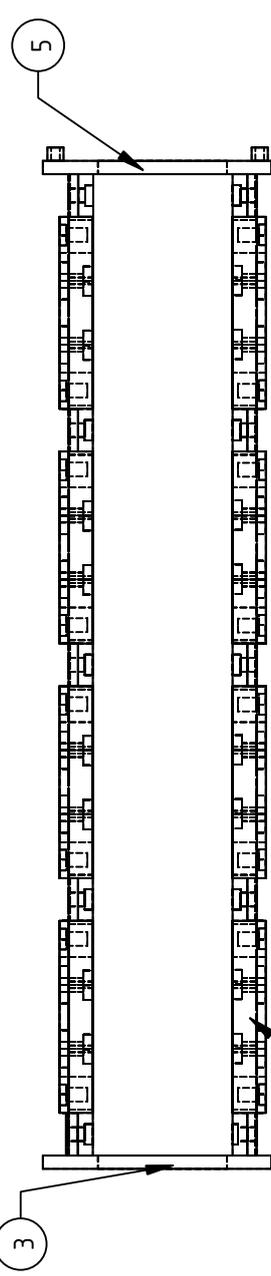


## **Apéndice M**

### **Planos de diseño de la batería y herramientas**

En este anexo, se presentan los planos de diseño de los módulos de batería, su carcasa y la carcasa del CAN2Serial.

H G U W Q A C M A



1	Plelina de Cobre
2	Plelina de Niquel
3	Tapa Lateral Izquierda
4	Soporte de Celda
5	Tapa Lateral Derecha

AUTHOR NAME: **Augusto Hernández**  
 DATE: **27/07/2021**  
 SUPERVISOR NAME: **Fernando Rosa**  
 DATE: **31/07/2021**  
 FORMAT: **A3**  
 SCALE: **1:2**  
 WEIGHT (kg): **ND**  
 VERSION: **1.0**

# Estructura de Bateria

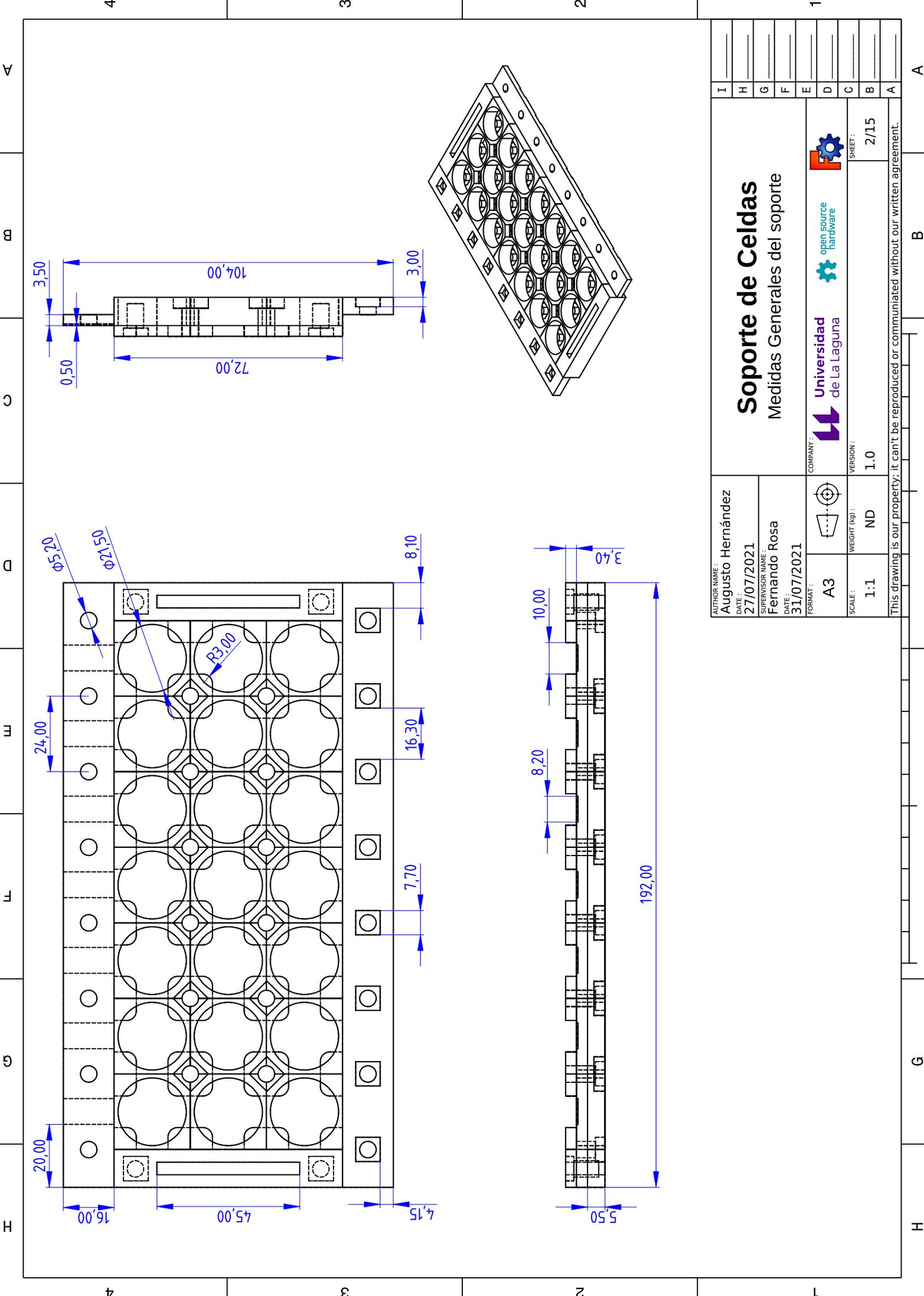
## Vista General de la Estructura

COMPANY: **Universidad de La Laguna**

SHEET: **1/15**

This drawing is our property; it can't be reproduced or communicated without our written agreement.

H G U W Q A C M A



AUTHOR NAME: <b>Augusto Hernández</b>		 COMPANY:  <b>Universidad de La Laguna</b>	 <b>open source hardware</b>	SHEET: <b>2/15</b>
DATE: <b>27/07/2021</b>				
SUPERVISOR NAME: <b>Fernando Rosa</b>		 WEIGHT (kg): <b>ND</b>	VERSION: <b>1.0</b>	SCALE: <b>1:1</b>
DATE: <b>31/07/2021</b>		This drawing is our property; it can't be reproduced or communicated without our written agreement.		
FORMAT: <b>A3</b>				

# Soporte de Celdas

## Medidas Generales del soporte

A B G H

4

3

2

1

4

3

2

1

A

B

C

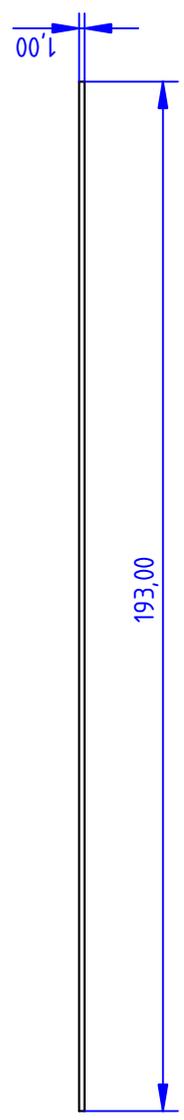
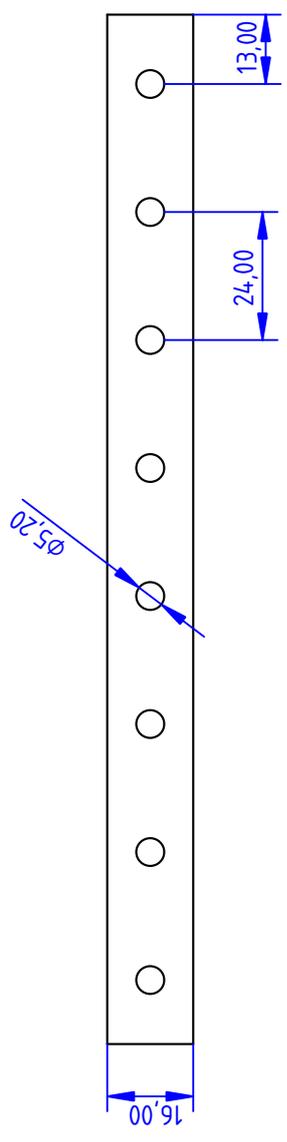
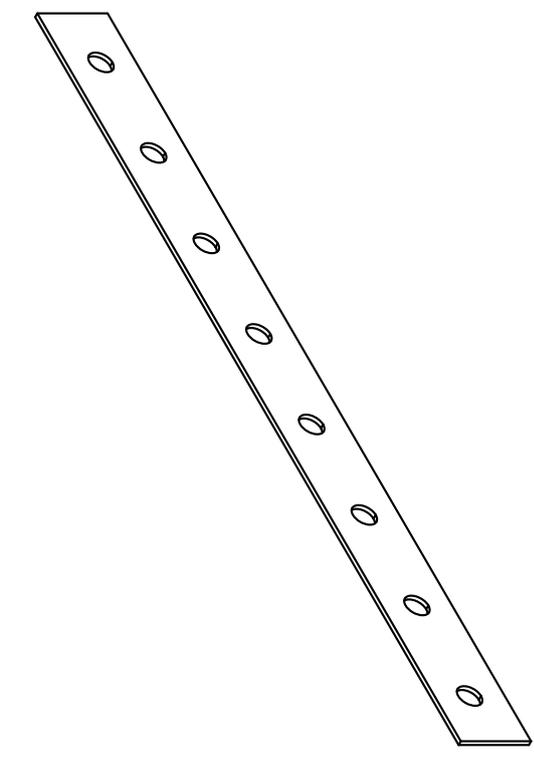
D

E

F

G

H



AUTHOR NAME:  
**Augusto Hernández**

DATE:  
**27/07/2021**

SUPERVISOR NAME:  
**Fernando Rosa**

DATE:  
**31/07/2021**

# Pletina de Cobre

## Medidas de Pletina de Cobre

COMPANY:  
**Universidad de La Laguna**

open source hardware

FORMAT:  
**A3**

WEIGHT (kg):  
**ND**

VERSION:  
**1.0**

SHEET:  
**3/15**

I
H
G
F
E
D
C
B
A

This drawing is our property; it can't be reproduced or communicated without our written agreement.

A

B

C

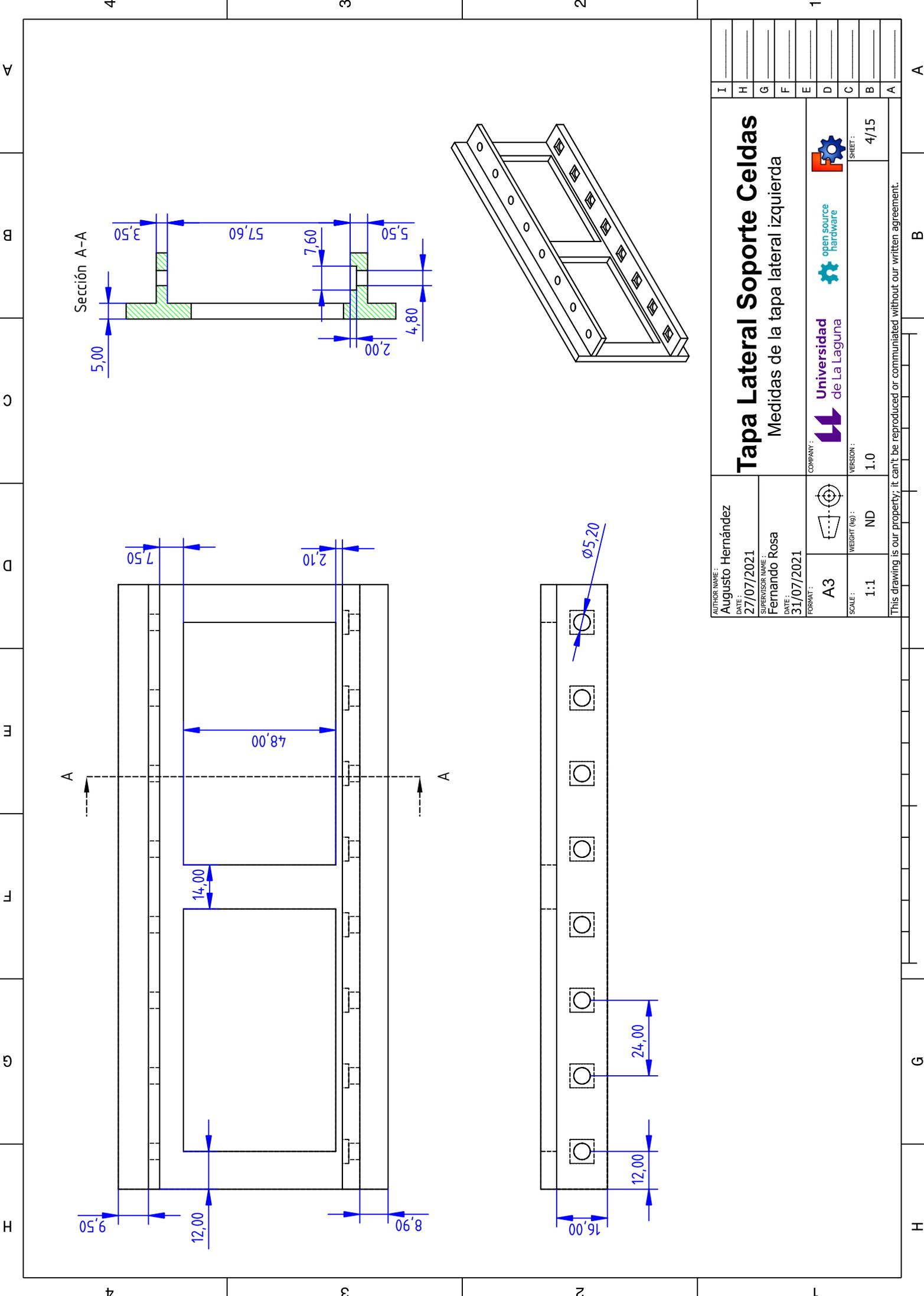
D

E

F

G

H



AUTHOR NAME:	Augusto Hernández
DATE:	27/07/2021
SUPERVISOR NAME:	Fernando Rosa
DATE:	31/07/2021
FORMAT:	A3
SCALE:	1:1
WEIGHT (kg):	ND
VERSION:	1.0

# Tapa Lateral Soporte Celdas

Medidas de la tapa lateral izquierda




SHEET: 4/15

I
H
G
F
E
D
C
B
A

This drawing is our property; it can't be reproduced or communicated without our written agreement.

4 3 2 1

A B

C

D

E

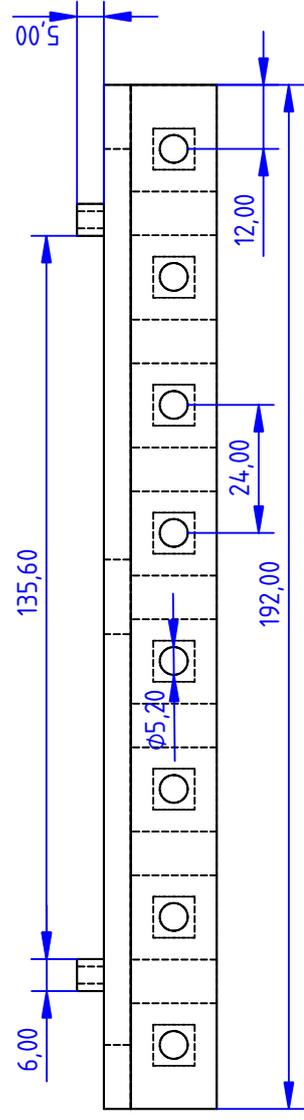
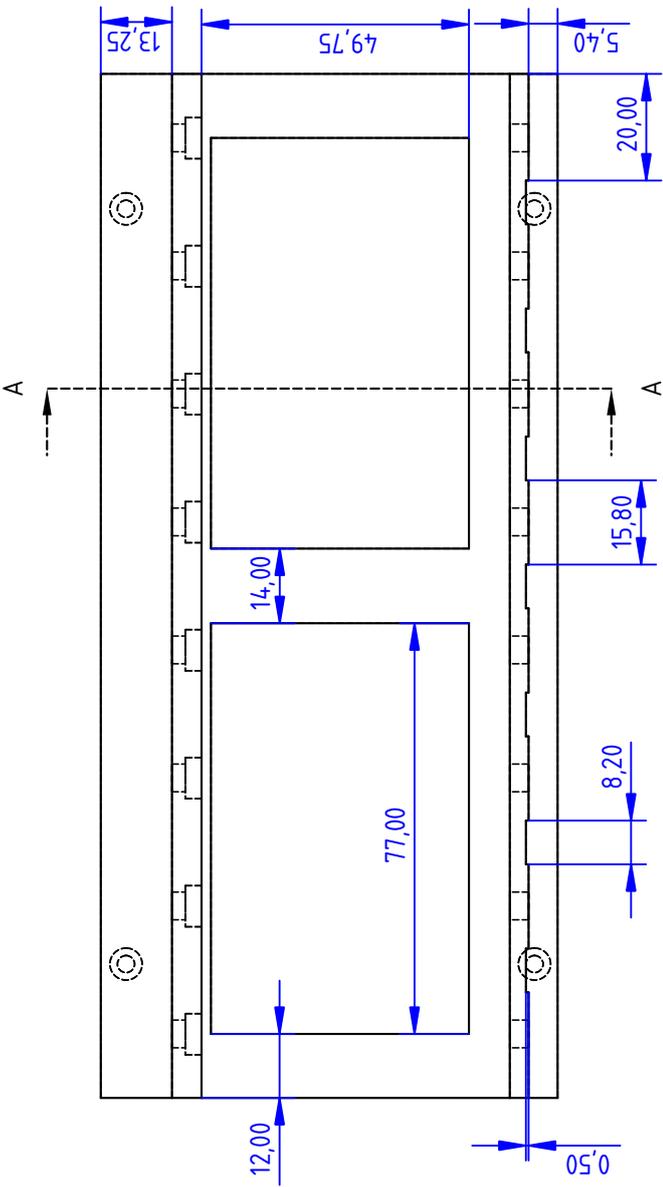
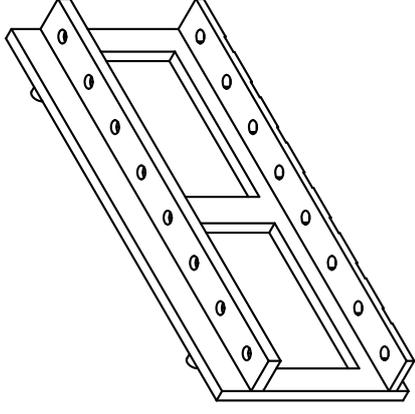
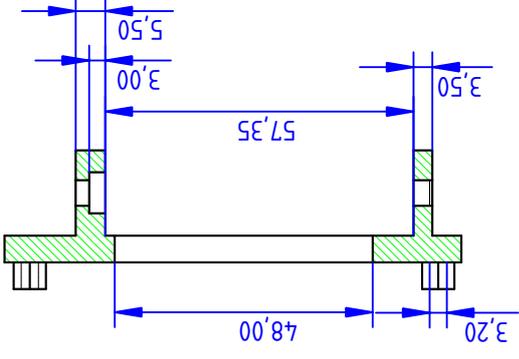
F

G

H

4 3 2 1

Sección A-A



AUTHOR NAME:	Augusto Hernández	COMPANY:	Universidad de La Laguna
DATE:	27/07/2021	open source hardware	
SUPERVISOR NAME:	Fernando Rosa		
DATE:	31/07/2021		
FORMAT:	A3		
SCALE:	1:1		
WEIGHT (kg):	ND		
VERSION:	1.0		
SHEET:	5/15		

# Tapa Lateral Derecha Celdas

Medidas de la Tapa Lateral Derecha

This drawing is our property; it can't be reproduced or communicated without our written agreement.

A B C D E F G H

H G U W D A B C O M A

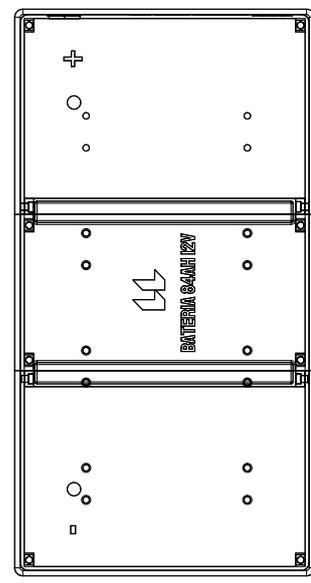
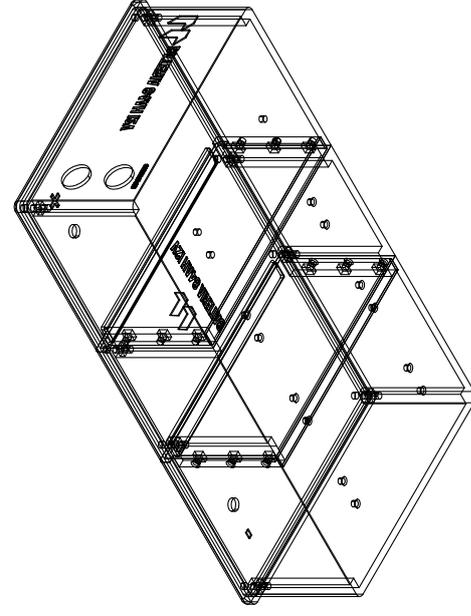
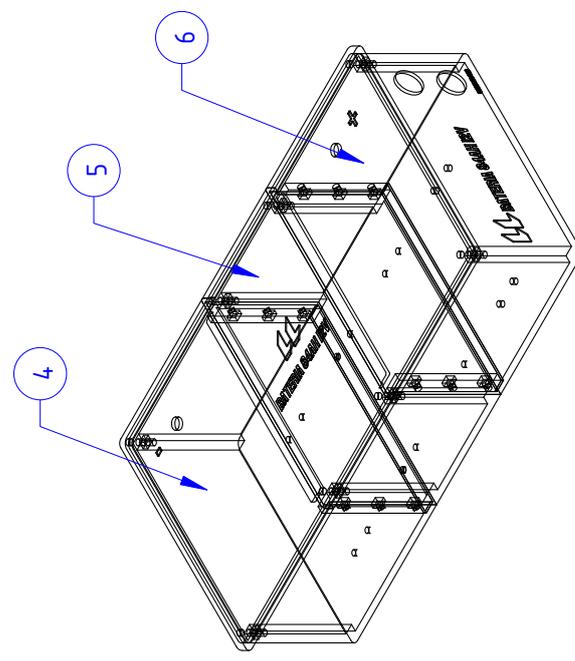
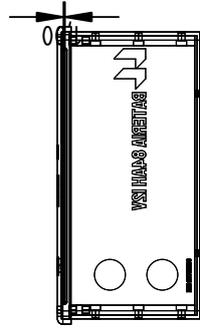
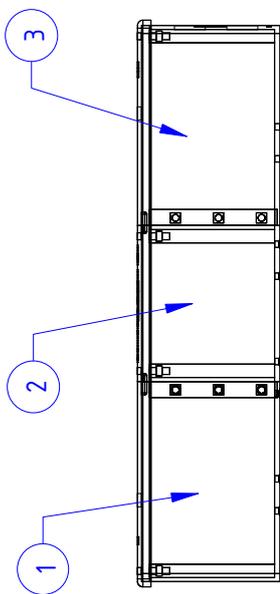
4 3 2 1

4

3

2

1



1	Base Izquierda Caja Batería
2	Base Central Caja Batería
3	Base Derecha Caja Batería
4	Tapas Izquierda Caja Batería
5	Tapas Central Caja Batería
6	Tapas Derecha Caja Batería

AUTHOR NAME:	Augusto Hernández
DATE:	27/07/2021
SUPERVISOR NAME:	Fernando Rosa
DATE:	31/07/2021
FORMAT:	A3
SCALE:	1:4
WEIGHT (kg):	ND
VERSION:	1.0
SHEET:	6/15

# Caja de Batería

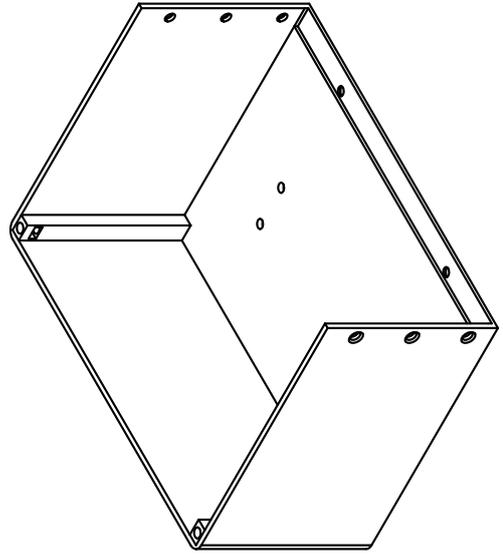
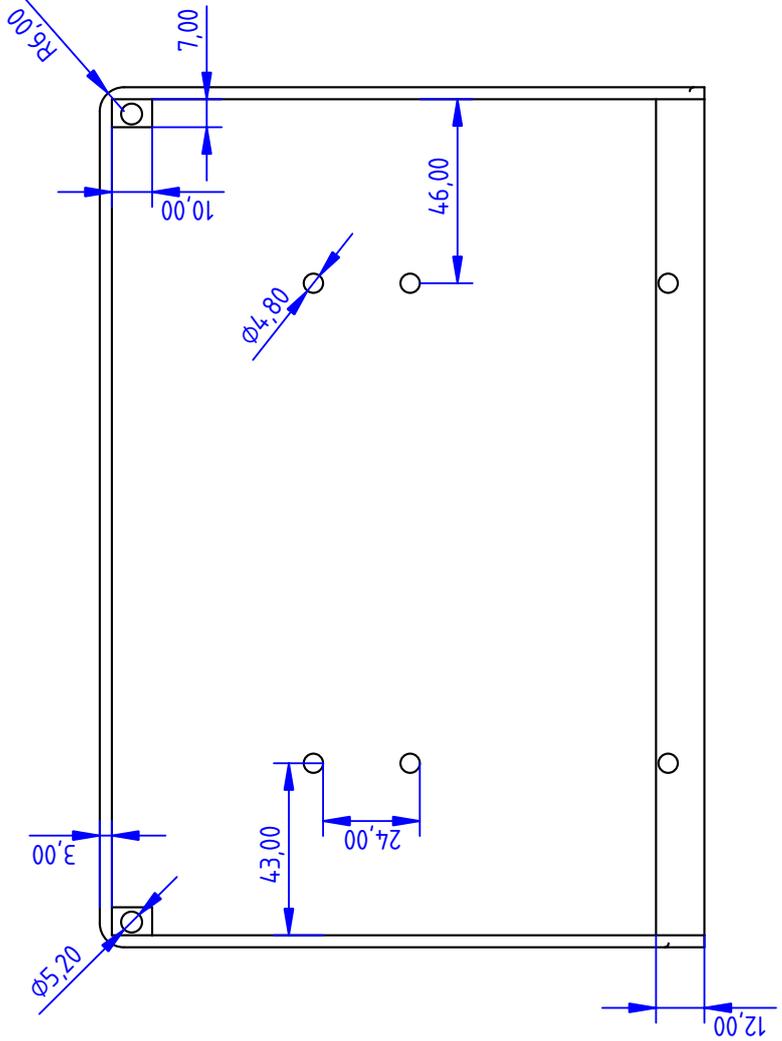
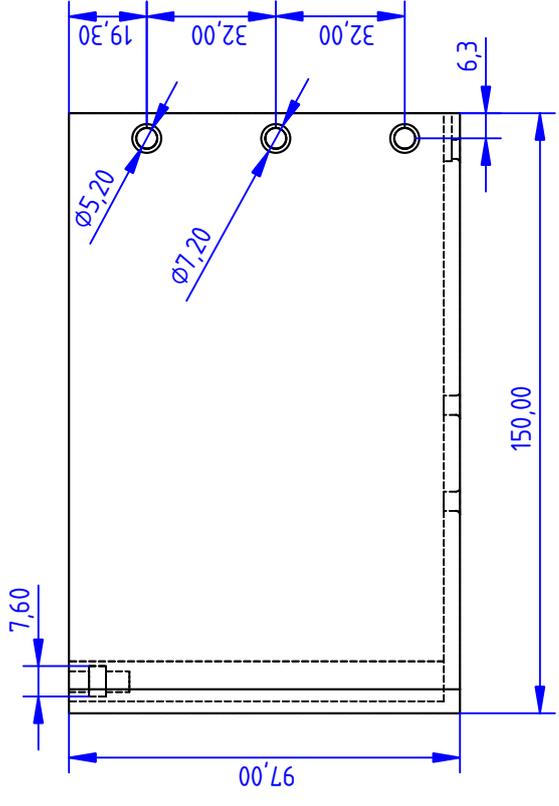
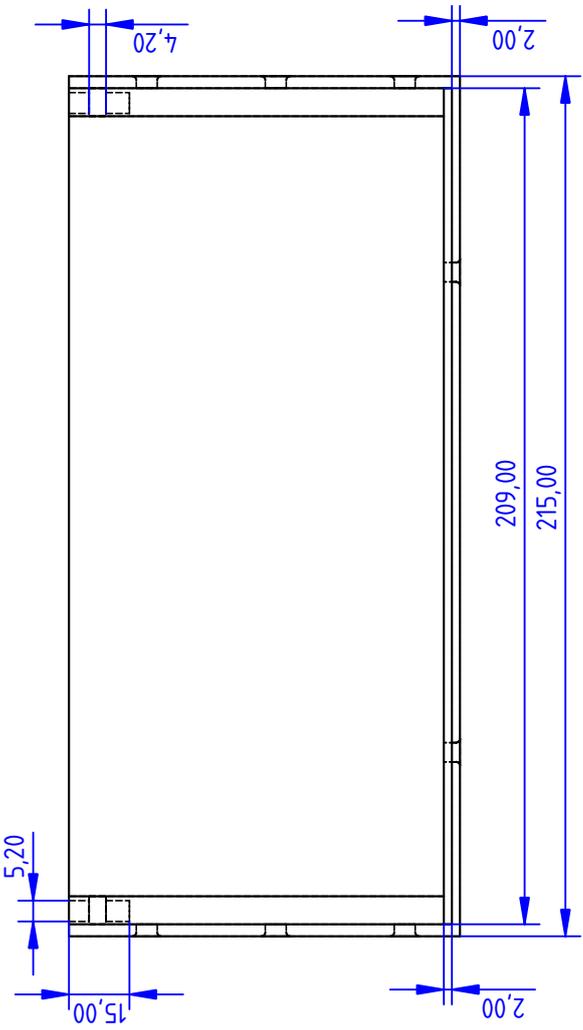
## Medidas Generales de la Caja de Batería

COMPANY:  
**Universidad de La Laguna**

open source hardware

This drawing is our property; it can't be reproduced or communicated without our written agreement.

A B C D E F G H I



AUTHOR NAME:	Augusto Hernández
DATE:	27/07/2021
SUPERVISOR NAME:	Fernando Rosa
DATE:	31/07/2021
FORMAT:	A3
WEIGHT (kg):	ND
VERSION:	1.0

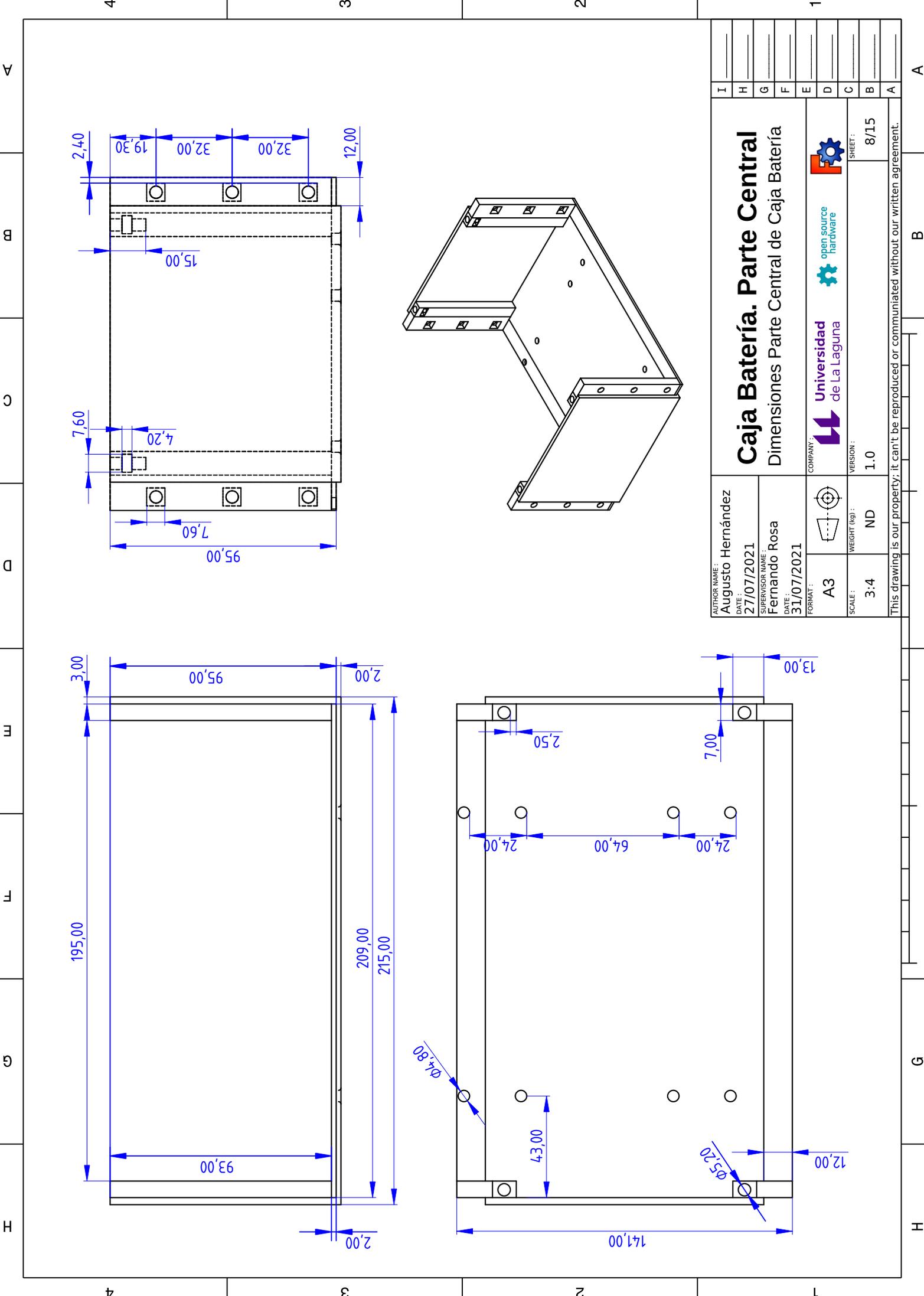
<b>Caja Bateria. Parte Izquierda</b> Dimensiones Parte Izquierda de Caja Bateria		 Universidad de La Laguna	 open source hardware	
This drawing is our property; it can't be reproduced or communicated without our written agreement.		SHEET: 7/15		

I
H
G
F
E
D
C
B
A

A B C D E F G H

4 3 2 1

A B C D E F G H



AUTHOR NAME:	Augusto Hernández	COMPANY:	Universidad de La Laguna
DATE:	27/07/2021	WEIGHT (kg):	ND
SUPERVISOR NAME:	Fernando Rosa	VERSION:	1.0
DATE:	31/07/2021	SCALE:	3:4
FORMAT:	A3		
<b>Caja Batería. Parte Central</b> Dimensiones Parte Central de Caja Batería		SHEET:	8/15

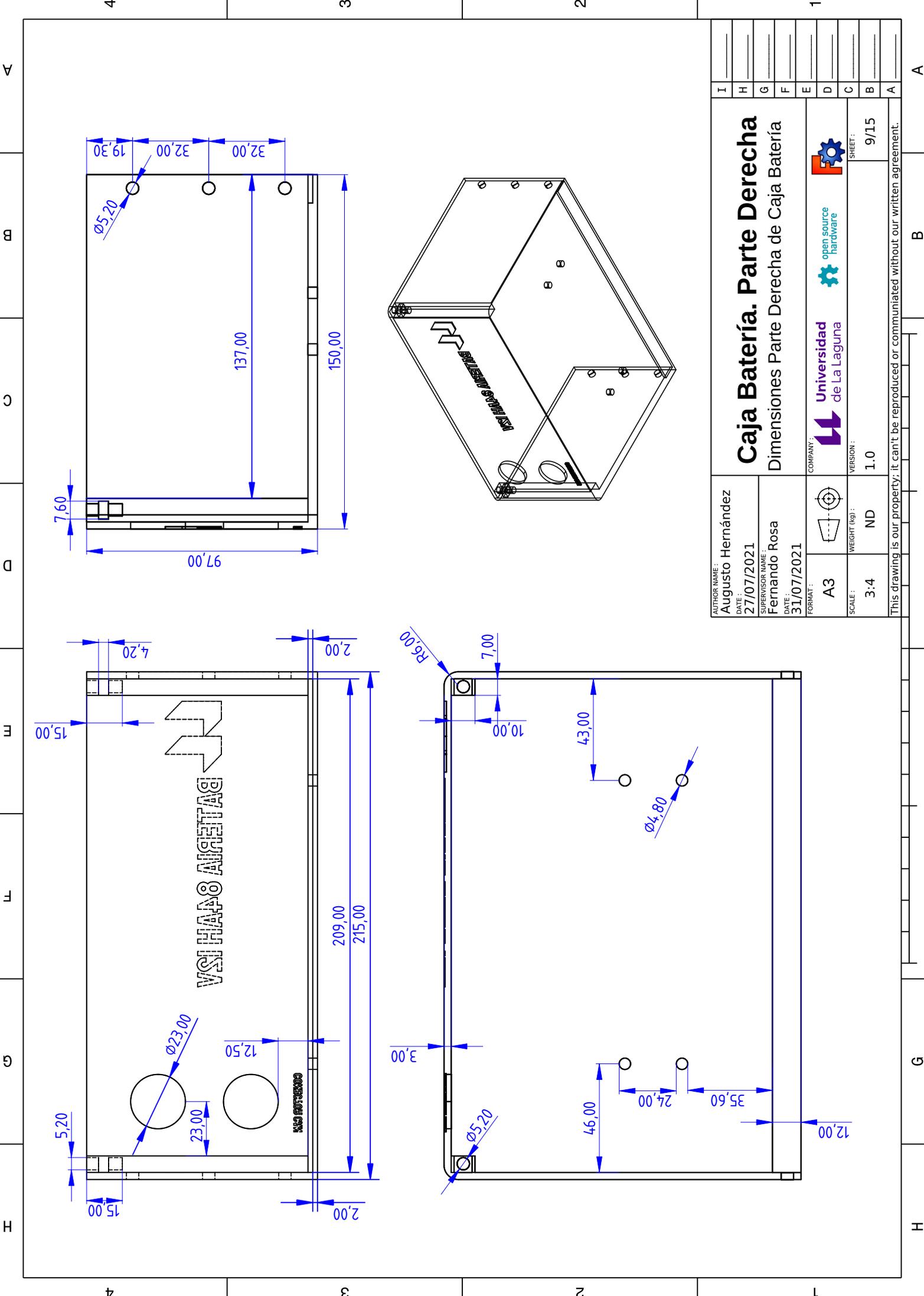
This drawing is our property; it can't be reproduced or communicated without our written agreement.

4 3 2 1

A B C D E F G H

4 3 2 1

A B C D E F G H



AUTHOR NAME: Augusto Hernández		I	
DATE: 27/07/2021		H	
SUPERVISOR NAME: Fernando Rosa		G	
DATE: 31/07/2021		F	
FORMAT: A3		E	
WEIGHT (kg): ND		D	
SCALE: 3:4		C	
VERSION: 1.0		B	
SHEET: 9/15		A	

# Caja Batería. Parte Derecha

## Dimensiones Parte Derecha de Caja Batería

COMPANY: **Universidad de La Laguna**

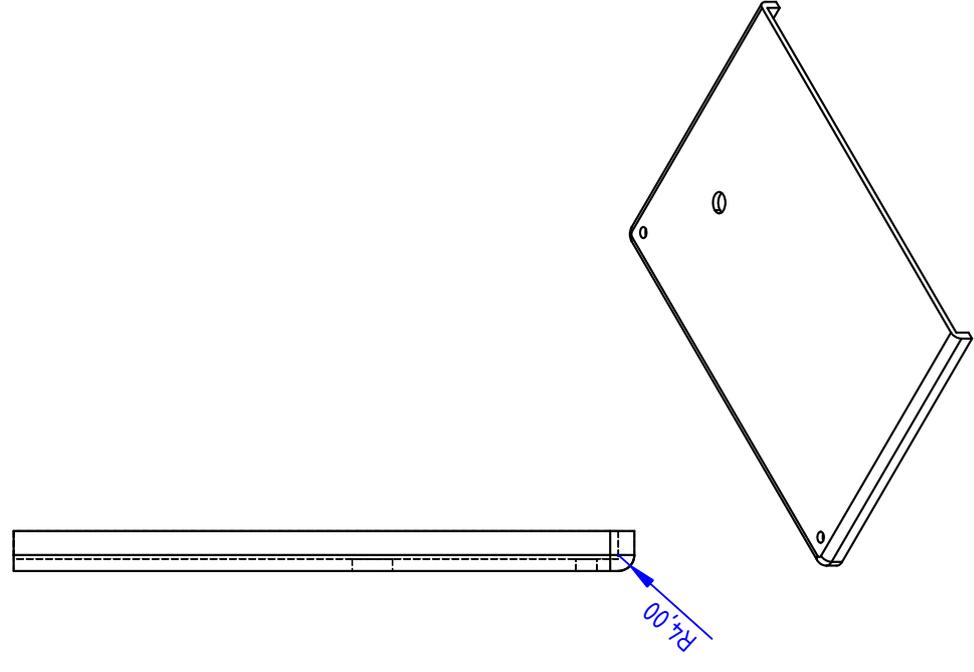
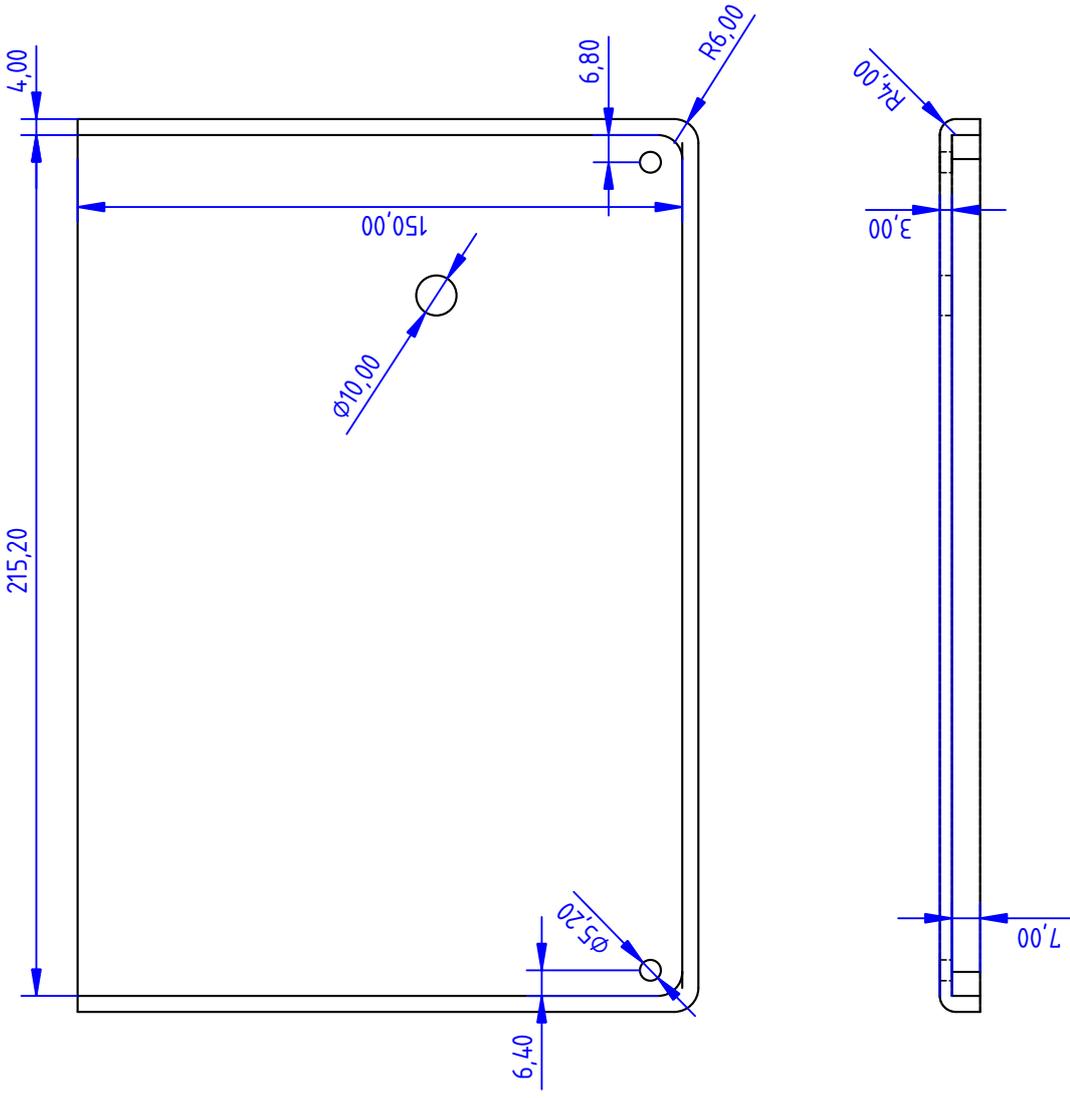
open source hardware

This drawing is our property; it can't be reproduced or communicated without our written agreement.

4 3 2 1

A B

H G F E D C B A

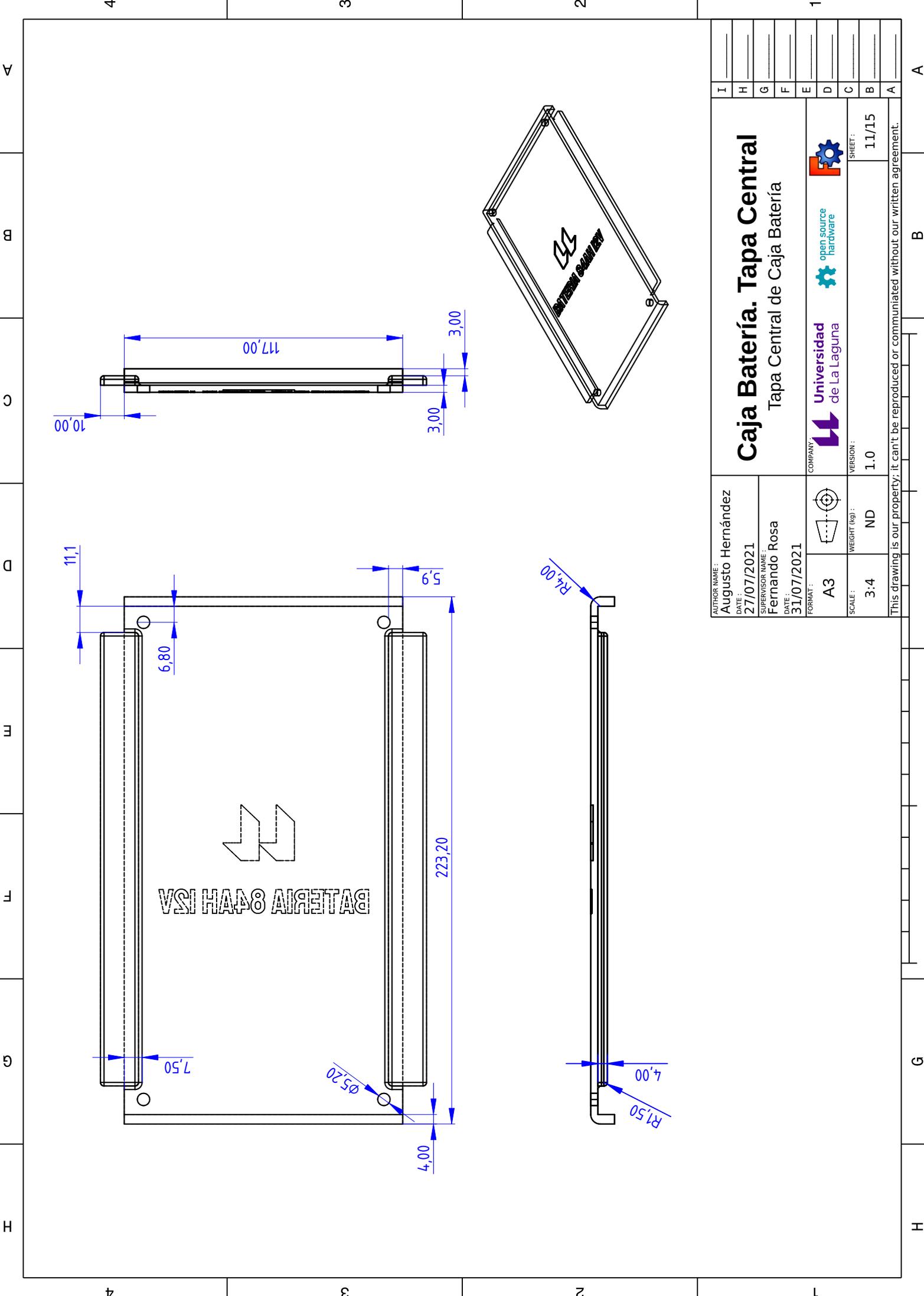


AUTHOR NAME:	Augusto Hernández	FORMAT:	A3
DATE:	27/07/2021	SCALE:	3:4
SUPERVISOR NAME:	Fernando Rosa	WEIGHT (kg):	ND
DATE:	31/07/2021	VERSION:	1.0
COMPANY:	 Universidad de La Laguna	 open source hardware	
SHEET:		10/15	

# Caja Batería. Tapa Izquierda

## Tapa Izquierda de Tapa Batería

This drawing is our property; it can't be reproduced or communicated without our written agreement.



AUTHOR NAME: <b>Augusto Hernández</b>		 A3	 WEIGHT (kg): ND	 COMPANY: <b>Universidad de La Laguna</b>	 open source hardware	 SHEET: 11/15
DATE: 27/07/2021	SUPERVISOR NAME: <b>Fernando Rosa</b>					
Tapa Central de Caja Bateria		This drawing is our property; it can't be reproduced or communicated without our written agreement.				

# Caja Bateria. Tapa Central

Tapa Central de Caja Bateria





A B G H

A B C D E F G H

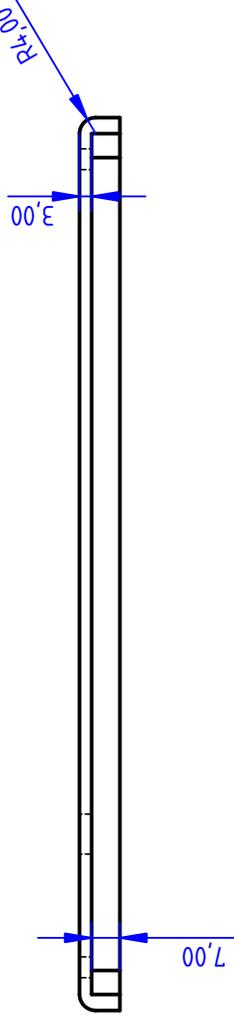
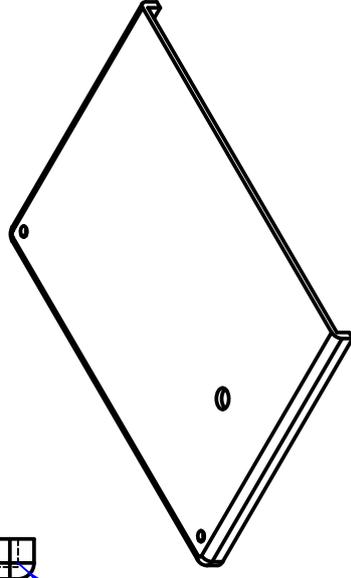
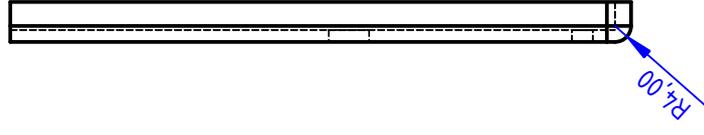
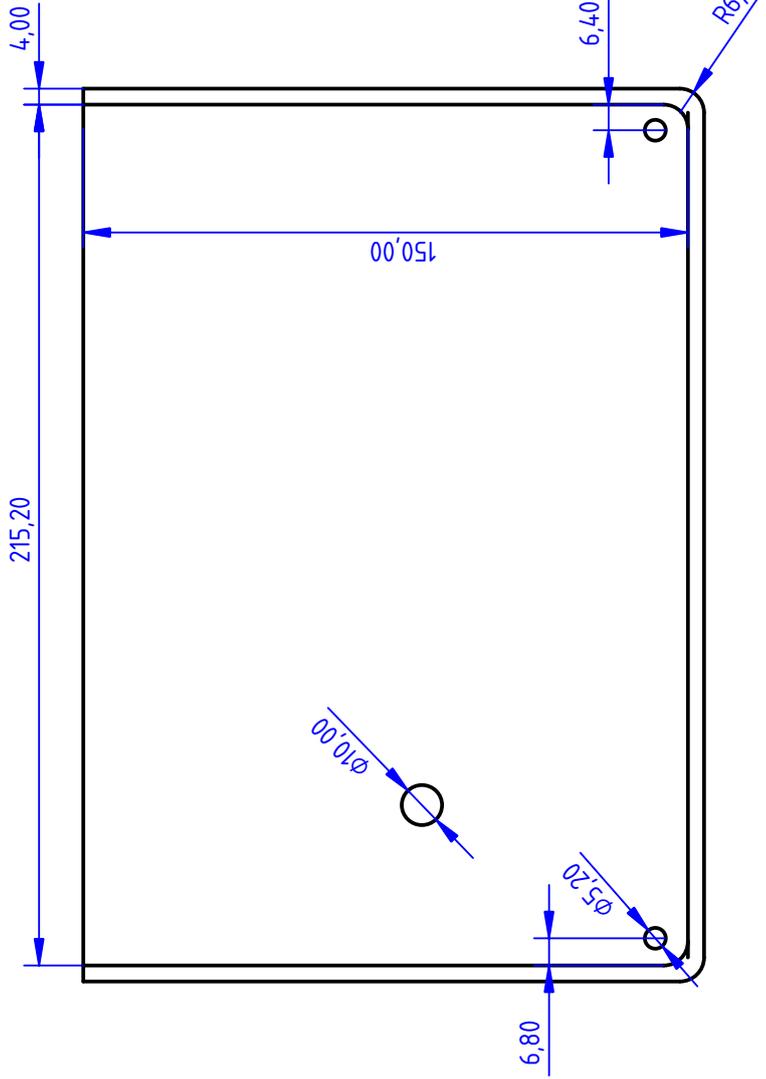
4 3 2 1

4 3 2 1

4 3 2 1

4 3 2 1

4 3 2 1



AUTHOR NAME:  
Augusto Hernández

DATE:  
27/07/2021

SUPERVISOR NAME:  
Fernando Rosa

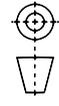
DATE:  
31/07/2021

FORMAT:  
A3

SCALE:  
3:4

WEIGHT (kg):  
ND

VERSION:  
1.0



# Caja Batería. Tapa Derecha

Tapa Derecha de Caja de Batería



COMPANY:  
Universidad  
de La Laguna

open source  
hardware

SHEET:  
12/15

VERSION:  
1.0

This drawing is our property; it can't be reproduced or communicated without our written agreement.

4

3

2

1

4

3

2

1

H

G

F

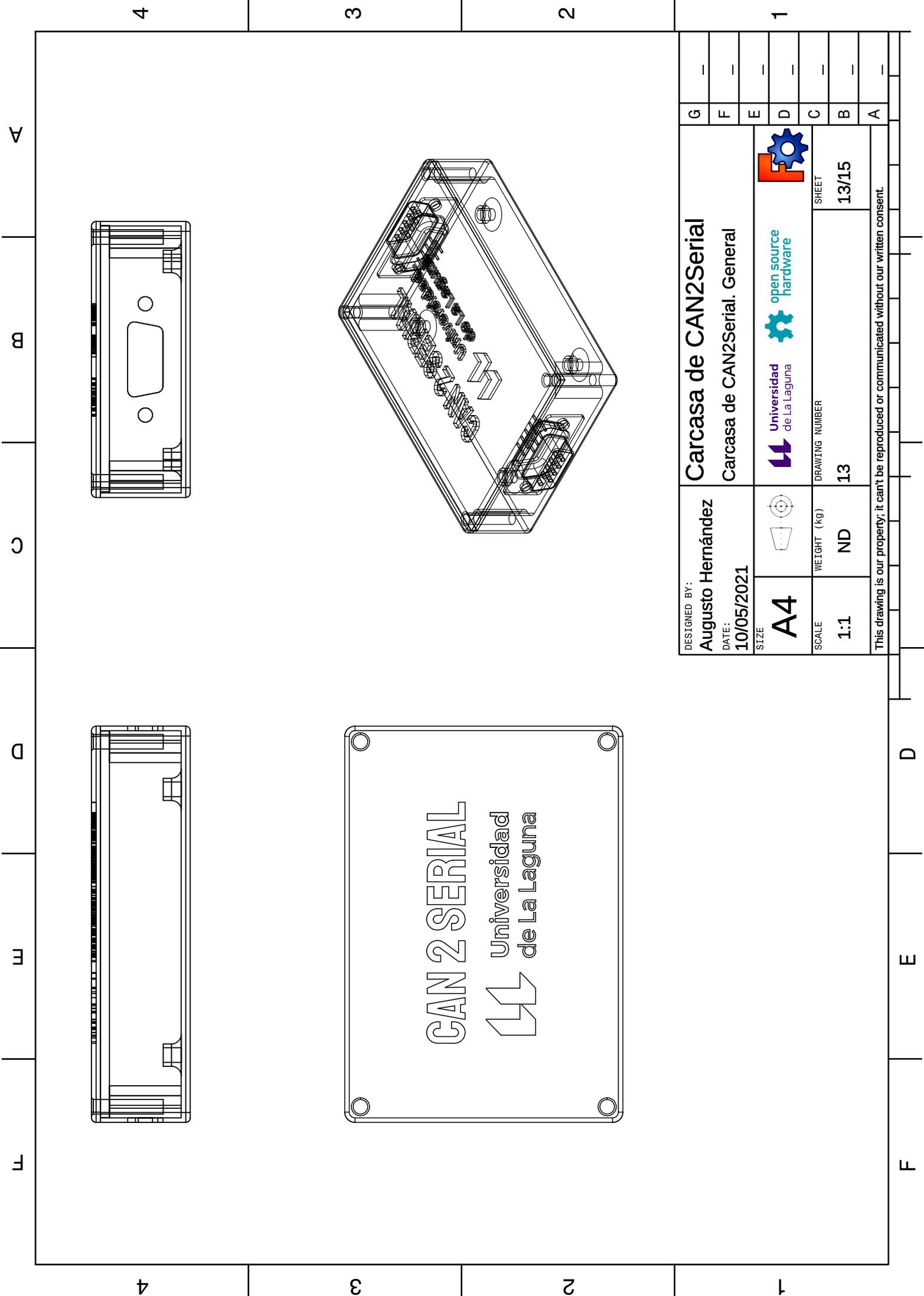
E

D

C

B

A



4

3

2

1

4

3

2

1

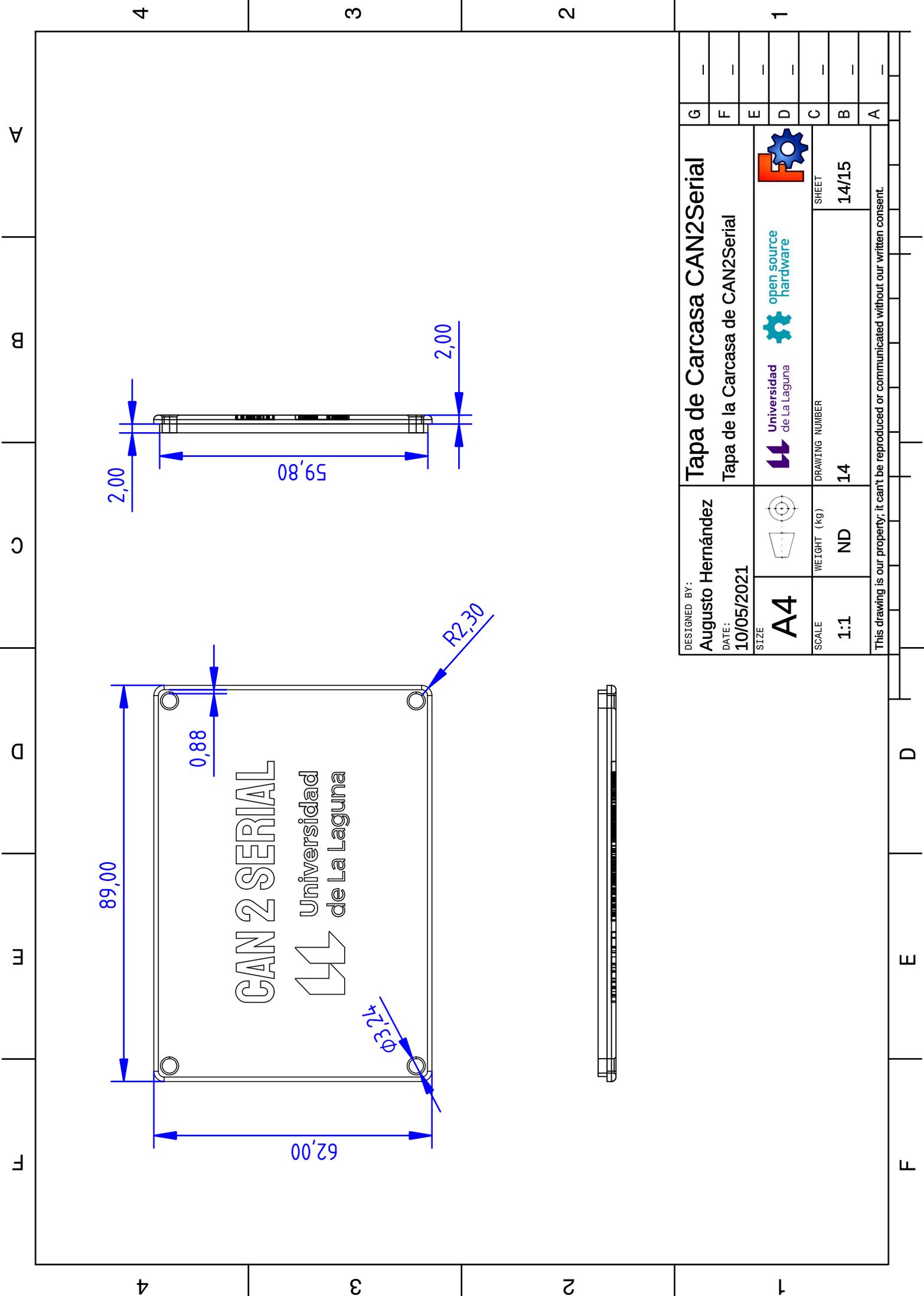
A B C D E F

DESIGNED BY: <b>Augusto Hernández</b>	<b>Carcasa de CAN2Serial</b>		G
DATE: <b>10/05/2021</b>	<b>Carcasa de CAN2Serial. General</b>		F
SIZE <b>A4</b>			E
SCALE <b>1:1</b>	WEIGHT (kg) <b>ND</b>	DRAWING NUMBER <b>13</b>	D
		SHEET <b>13/15</b>	C
This drawing is our property; it can't be reproduced or communicated without our written consent.			B
			A

F

E

D



DESIGNED BY: <b>Augusto Hernández</b>			
DATE: <b>10/05/2021</b>			
SIZE <b>A4</b>	WEIGHT (kg) <b>ND</b>	DRAWING NUMBER <b>14</b>	SHEET <b>14/15</b>

**Tapa de Carcasa CAN2Serial**  
Tapa de la Carcasa de CAN2Serial

This drawing is our property; it can't be reproduced or communicated without our written consent.

Grid labels: A, B, C, D, E, F (horizontal); 1, 2, 3, 4 (vertical)

